



**UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO
CENTRO UNIVERSITARIO UAEM TEXCOCO**

“Sistema para la Administración de Capital Humano de la
Secretaría de Administración Tributaria”

T E S I S

QUE PARA OBTENER EL TÍTULO DE
INFORMATICA ADMINISTRATIVA

P R E S E N T A

STEPHANIE NEGLAHED
CASTILLO CAMACHO

FABIOLA ZAVALA CRUZ

DIRECTOR:

DR. EN C. ADRIAN TRUEBA ESPINOSA

REVISORES:

DRA. REBECA TEJA GUTIÉRREZ

M. EN C, JOSÉ SERGIO RUIZ CASTILLA

M. EN C. C. ÁNGEL RAFAEL QUINTOS
RAMÍREZ

El éxito no se logra sólo con cualidades especiales. Es sobre todo un trabajo de constancia, de método y de organización.

(J.P. Sergent)

Agradecimientos

A mi Familia:

A quien jamás encontraré la forma de agradecer el que me haya brindado su mano en las derrotas y logros de mi vida. Agradecemos su apoyo, comprensión y confianza esperando que comprendan que mis logros son también tuyos e inspirados en tí, hago de este un triunfo y quiero compartirlo por siempre contigo.

A mi Director y Revisores:

Por su experiencia, paciencia, guía y apoyo. Este presente simboliza mi gratitud por toda la responsabilidad e invaluable ayuda que siempre nos han proporcionado, porque gracias a sus consejos, hemos llegado a realizar una de nuestras grandes metas.

A mis Amigos:

Por el apoyo moral que siempre he recibido de ustedes y con el cual he logrado culminar mi esfuerzo, terminando así mi carrera profesional, quiero expresar un profundo agradecimiento a quienes con su ayuda y comprensión me alentaron a lograr esta hermosa realidad.

Dedicatoria

Este proyecto lo dedizamos a nuestro esfuerzo, tolerancia y a la constancia que tuvimos a largo de este logro que conseguimos juntas. A Fabiola y Stéphanie.

Gracias

CONTENIDO

I.- INTRODUCCIÓN	9
II.- PLANTEAMIENTO DEL PROBLEMA	11
III.- JUSTIFICACIÓN	12
IV.- OBJETIVOS.....	13
4.1.- Objetivo general:.....	13
4.2.- Objetivos específicos:	13
V.- MARCO TEÓRICO	14
5.1.- Antecedentes de los sistemas de información para la administración del capital humano.....	14
5.1.1.- Capital Humano.....	14
5.1.2.- Administración de Recursos Humanos (ARH) y los sistemas de información	18
5.1.3.- Sistemas de información para la administración de recursos humanos	19
5.1.4.- Influencia de la Tecnología en la Administración de Recursos Humanos (ARH).....	28
5.2.- Desarrollo de un sistema de información.....	32
5.2.1.- Características de un sistema de información	32
5.2.2.- Categorías de sistemas de información.....	33
5.2.3.- Estrategias para el desarrollo de sistemas	34
5.2.4.- Análisis de Requerimientos	35
5.2.5.- Requerimientos del sistema.....	35
5.2.6.- Lineamientos al definir los requerimientos.....	38
5.2.7.- Características de los requerimientos.....	38
5.2.8.- Dificultades para definir los requerimientos	39
5.2.9.- Tipos de Requerimientos	41
5.2.10.- Técnicas y herramientas utilizadas en la Ingeniería de Requerimientos (IR).....	42

5.3.- Ingeniería de Software	44
5.3.1.- Aplicaciones del Software	46
5.3.2.- Proceso de Software	48
5.3.3.- Modelos de proceso de software	50
5.3.3.1.- Modelo de Construcción de Prototipos	52
5.3.3.2.- Modelos DRA.....	53
5.3.3.3.- Modelo Incremental	55
5.3.3.4.- Modelo Espiral	56
5.4.- Modelado (UML)	59
5.4.1.- Historia (UML).....	61
5.4.2.- UML	61
5.4.3.- Diagramas UML.....	62
5.4.4.- Proceso Unificado de Desarrollo	66
5.4.5.- Casos de uso	67
5.5.- Base de datos	76
5.5.1.- Ventajas de las bases de datos	79
5.5.2- Modelo relacional.....	79
5.5.3.- Tablas	81
5.5.4.- Esquema entidad relación	83
5.5.5.- Relaciones	83
5.5.6.- Normalización	85
5.5.7.- Sistemas manejadores de bases de datos relacionales	87
5.5.8.- Definición y funciones de un (DBA)	88
5.5.9.- SMDDB comerciales y de software libre	89
5.6.- SQL.....	89
5.6.1.- Breve historial SQL.....	89
5.6.2.- Lenguaje SQL.....	91
5.6.3.- Características de SQL.....	91
5.6.4.- Componentes de SQL	92
5.6.4.1.- Comandos SQL	92
5.6.4.2.- Cláusulas	94

5.6.4.3- Operadores y condiciones	95
5.6.4.4.- Funciones de agregado	97
5.6.5.- Consulta.....	99
5.6.6.- Cláusula WHERE	101
5.6.7.- Cláusula ORDER BY	102
5.6.8.- Cláusula GROUP BY	103
5.6.9.- Cláusula HAVING	103
VI. METODOLOGÍA	104
6.1.- Pasos Metodológicos.....	104
6.1.1.- Estructuración de la entrevista para conocer las necesidades del sistema y delimitar el alcance del sistema de información	104
6.1.2.- Determinar las reglas de negocios que reflejen las necesidades de los usuarios.	104
6.1.3.- Identificar datos y reportes.....	106
6.1.4.- Diseño del modelo relacional de la base de datos.....	111
6.1.5.- Implementación física de la base de datos en MySQL.	111
6.1.6.- Análisis y diseño del sistema de información en UML, casos de uso..	112
6.1.7.- Diseño de vistas del sistema.	115
6.1.8.- Codificar o Programar los algoritmos previamente diseñados.....	115
6.1.9.- Implementación del Sistema de Información	116
VII.- RESULTADOS.....	119
VIII.- DISCUSIÓN.....	130
IX.- CONCLUSIÓN	140
X.- RECOMENDACIONES.....	141
XII.- BIBLIOGRAFÍA	142
XIII. ANEXOS	150

INDICE DE TABLAS

Tabla 1.- Cualidades de la información	20
Tabla 2.- Parámetros de los sistemas	22
Tabla 3.- Sistemas de información de recursos humanos en el mercado	29
Tabla 4.- Lineamientos de los requerimientos	38
Tabla 5.- Características de los requerimientos	39
Tabla 6.- Comparación de relaciones Desarrollador-Usuario y Usuario-Desarrollador	40
Tabla 7.- Tipos de requerimientos	41
Tabla 8.- Aplicaciones del software	46
Tabla 9.- Proceso de software (Ejemplos).....	52
Tabla 10.- Tipos de “Actores”	70
Tabla 11.- Tipos de bases de datos	77
Tabla 12.- Definiciones y conceptos de bases de datos.....	78
Tabla 13.- Evolución del modelo relacional	79
Tabla 14.- Tipos de relaciones entre tablas.....	84
Tabla 15.- Ventajas y Desventajas de un SDBD.....	88
Tabla 16.- Evolución de SQL.....	90
Tabla 17.- Comandos DML	93
Tabla 18.- Clausulas SQL	94
Tabla 19.- Operadores lógicos SQL	95
Tabla 20.- Operadores de comparación SQL.....	96
Tabla 21.- Operadores de caracteres SQL.....	96
Tabla 22.- Operadores matemáticos SQL	97
Tabla 23.- Funciones de agregados SQL.....	98
Tabla 24.- Funciones de caracteres SQL	98
Tabla 25.- Consultas con predicados	100
Tabla 26.- Diccionario de Datos.	107
Tabla 27.- Sistema de información de recursos humanos SAT System.....	119
Tabla 28.- Iconos del sistema.....	129

ÍNDICE DE FIGURAS

Figura 1 Actividades de un sistema de información.	23
Figura 2 Porcentaje de empresas que emplean equipos de cómputo en procesos administrativos	24
Figura 3 Dificultades de los requerimientos.....	40
Figura 4 Modelo de construcción de prototipos.....	53
Figura 5 Modelo DRA	54
Figura 6 Modelo Incremental.....	55
Figura 7 Modelo espiral.....	56
Figura 8 Símbolo de un actor	69
Figura 9 Asociación Casos de Uso.....	71
Figura 10 Extensión Casos de Uso	72
Figura 11 Usos Casos de Uso.....	73
Figura 12 Dependencia Casos de Uso.....	74
Figura 13 Herencia Casos de Uso.....	75
Figura 14 Partes de una tabla	81
Figura 15 Ejemplo de Llave primaria	82
Figura 16 Ejemplo de Llave Foránea.....	83
Figura 17 Diagrama de flujo del sistema de Recursos Humanos.....	106
Figura 18 Tablas y catálogos realizadas en MySQL	111
Figura 19 Mysql Fuente MySQL.....	112
Figura 20 Casos de uso Guardar.	112
Figura 21 Casos de uso Buscar.	113
Figura 22 Casos de uso Modificar.....	113
Figura 23 Casos de uso Generación de Reportes.....	114
Figura 24 Casos de uso exportación de reportes.....	114
Figura 25 Vista principal del sistema en Visual Basic 6.0.....	115
Figura 26 Visual Basic 6.0. Fuente :Visual Basic 6.0	116
Figura 27 formularios del sistema.	117
Figura 28 Reporte del sistema en Visual Basic 6.0..	117

Figura 29 Base de Datos Relacional del sistema para administración de Recursos Humano.....	118
Figura 30 Login..	120
Figura 31 Pantalla principal.	120
Figura 32 Catálogos..	121
Figura 33 Catálogo de Movimientos.....	121
Figura 34 Catálogo de nivel.....	122
Figura 35 Catálogo de rango.....	122
Figura 36 Catálogo de ubicación.	123
Figura 37 Catálogo Causa de la baja.	123
Figura 38 Catálogo área.....	124
Figura 39 Control del empleado..	124
Figura 40 Exportar el historial del empleado a Excel.....	125
Figura 41 Reportes.....	126
Figura 42 Reportes de movimiento.	126
Figura 43 Generar registros a excel.	127
Figura 44 Reporte de bajas..	127
Figura 45 Seguridad..	128
Figura 46 Generación de reportes	131
Figura 47 Reportes de movimientos a nivel Detalle.	132
Figura 48 Reportes de movimientos a nivel Resumen.	133
Figura 49 Reportes de bajas a nivel Detalle.....	134
Figura 50 Reportes de bajas a nivel Resumen.....	134
Figura 51 Reportes de plaza a nivel Detalle.....	135
Figura 52 Reportes de plaza a nivel Resumen	136
Figura 53 Historial del empleado	137
Figura 54 Exportación de la información a Excel.	137

I.- INTRODUCCIÓN

Si se desea una administración de recursos humanos eficiente y eficaz, se debe trabajar en conjunto es decir implementar un sistema de información para la administración de recursos humanos que se ajuste a las necesidades de la organización y contar con el personal capacitado para el manejo adecuado del sistema, ya que se cree que si el personal que opere el sistema no utiliza al máximo el potencial del mismo no se contara con los resultados deseados [28].

Las organizaciones en cada una de las áreas tienen la necesidad de tomar decisiones, desde la más insignificante hasta aquella que tiene que ver con su existencia misma. Con tal responsabilidad se necesita una base de datos lo más certera posible sobre la cual descansen esas decisiones [25].

Tal es el caso del departamento de Recursos Humanos del SAT (Secretaria de Administración Tributaria), el cual cuenta con una plantilla de personal muy extensa y dinámica ya que los servidores públicos están cambiando constantemente. El departamento está encargado de llevar el registro, control, actualización y primordialmente autenticidad de la información de cada uno de los servidores públicos de sus 11 Administraciones Centrales. El método que se utiliza es manual y requiere demasiado tiempo para su control lo que atrasa la búsqueda de información, reportes, actualización y toma de decisiones en tiempo real.

Por lo anterior, el objetivo de este trabajo es desarrollar un Sistema de Información para el control de la administración de dicha institución, un Sistema que beneficie en primera instancia tener la información de forma inmediata y actual de cada empleado, manteniendo actualizada la estructura organizacional.

Este proyecto iniciará con el análisis de los procesos para la identificación de los requerimientos, que permitan el planteamiento de un sistema de información, a partir de los procesos administrativos movimientos de personal, a fin de diseñar una

base de datos, que junto con la programación de algoritmos implementados en Visual Basic 6.0 para la ejecución de reportes, se tendrá como producto final un Sistema de Administración de Recursos Humanos que se validará a través de la operación de algunos integrantes del departamento, a fin de mejorar el sistema tal como lo señala el método de espiral para implementar sistemas de información.

II.- PLANTEAMIENTO DEL PROBLEMA

El SAT (Secretaría de Administración Tributaria) tiene contratada una plantilla de personal muy grande y es muy dinámica ya que los servidores públicos están cambiando constantemente de cargo y área. Se debe llevar el registro del personal de 11 áreas dependientes del SAT, lo cual conlleva a la generación de deficiencias en el control, ya que el manejo se hace de manera manual. Los servicios que ofrece el área de Recursos humanos es: control de ingreso, reingreso y egresos. Todo movimiento relacionado con el empleado se maneja mediante documentos específicos dependiendo del trámite solicitado, (atenta nota, constancias y formatos SMP-01). Se controlan mediante el manejo de archivos en papel. Lo cual ocasiona inconsistencia en la administración de personal lo que trae como consecuencia:

- Pérdida de tiempo en buscar la información requerida.
- Entrega de reportes quincenales, mensuales y anuales a destiempo.
- Autenticidad del historial de cada empleado.
- Mantener actualizada la estructura de cada área.
- Falta de información en tiempo real.
- Pago de nómina con atraso o incorrecta por no registra un cambio en tiempo y forma.
- Reingreso de personal no deseado, por no tener un historial adecuado.
- Toma de decisiones sin tener información en tiempo real.

III.- JUSTIFICACIÓN

Las empresas del sector público y privado que manejan muchos empleados con movimientos constantes y que desean registrar los cambios de área y puesto con un registro histórico, es deseable que cuenten con un sistema automatizado capaz de proporcionar información de manera instantánea y en tiempo real. Los beneficios se tienen son:

- En primera instancia tener la información de forma Inmediata.
- Contar con un historial real y actual de cada empleado.
- Tener actualizada la estructura organizacional de forma jerárquica.
- Ofrecer informes actualizados.

El manejo adecuado del Capital Humano trae como consecuencia un mejor control en la administración del personal y por ende se tendrá un buen desempeño en eficiencia a un bajo costo, ya que se destinaría personal a actividades sustantivas en este caso del SAT y con ello mejorar los servicios al usuario con la reducción de tiempos en procesos de registro a contribuyentes.

IV.- OBJETIVOS

4.1.- Objetivo general:

Desarrollar un Sistema de Información para el control de Capital Humano de la Secretaria de Administración Tributaria.

4.2.- Objetivos específicos:

- Determinar las necesidades del sistema de información para delimitar el alcance del trabajo planteado.
- Establecer las reglas de negocios del sistema de información para cumplir con los requerimientos del usuario.
- Diseñar el modelo de bases de datos para integrar el sistema de información.
- Planteamiento de los algoritmos del sistema de información para cumplir con las reglas de negocios.
- Programación de los algoritmos para implementar el sistema de información.
- Estructuración del sistema de información para su operatividad.

V.- MARCO TEÓRICO

5.1.- Antecedentes de los sistemas de información para la administración del capital humano.

5.1.1.- Capital Humano.

El pensamiento del hombre ha evolucionado tan rápidamente, que, en los últimos cincuenta años, la importancia y el impacto que poseen las personas en las organizaciones han evolucionado conceptualmente y valorándose de diferente manera, antes se consideraba al hombre como una simple pieza sin valor, ahora se ha convertido en la parte esencial de la institución. Para explicar esta evolución [29], destaca cronológicamente lo siguiente: en los 40's, se introdujeron por primera vez, prestaciones al salario monetario, tales como, servicios médicos y dotación de viviendas. En los 50's y 60's se promueven los sindicatos para la defensa de los derechos laborales de los trabajadores. En los 70's, se hace necesario contratar personal especializado por lo que se requiere de profesionales universitarios especializados en el área, los relacionistas industriales, se tecnifican los procesos: particularmente reclutamiento y selección, inducción y administración de sueldos y salarios. Se formalizan los procesos de la administración de personas. Ya en los 80's, se hizo necesario contratar personas con preparación gerencial (consultores, académicos, otros), considerando a la Gerencia de los Recursos Humanos como el "factor estratégico futuro" o la "nueva arma competitiva", para ello se hizo necesario identificar personas con capacidades relacionados con la productividad, la cultura organizacional o la filosofía gerencial, ya que se consideró como un factor determinante para el éxito de una organización o empresa. En los 90's, se produce cambios acelerados ante los avances tecnológicos, esta década se caracteriza como la era del conocimiento. En el siglo XXI, con el continuo avance de la información y la tecnología aparece una nueva realidad. Las organizaciones y empresas están montadas en la dinámica de tiempos de cambios requeridos por las tecnologías, crecimiento rentable y exigencias por parte de los clientes, colocando

en el centro de la escena, la competencia de la fuerza laboral y las capacidades de la empresa.

En los últimos cinco años (2005-2010) aparece otro concepto relacionado con las funciones de recursos humanos, el cual, todavía se pone de manifiesto en la práctica empresarial diaria en las organizaciones, y sobre el cual, varios autores han realizado valoraciones y aportes teóricos importantes. El concepto es “Gestión de Recursos Humanos”; en él se considera una organización de éxito, de carácter estratégico y con un organización cimentada en el al proceso de formación de personal [29].

Desde el 2001 al 2013 las empresas han acuñado el término capital humano, el cual se define, como el cumulo de conocimientos, experiencias y saberes de los recursos humanos en una organización [3].

En los últimos tiempos emergen con fuerza los llamados “activos intangibles” o de una manera más general, el capital intelectual. Existe una gran cantidad de criterios sobre el capital intelectual, el cual está formado por [36].

- (1) Capital humano. El cual incluye todas las capacidades individuales, los conocimientos y destrezas de los empleados.
- (2) Capital estructural. Se entiende el conjunto de los equipos, programas, bases de datos, patentes, marcas y estructura organizacional. También incluye las relaciones con los clientes.
- (3) Capital relacional. Es la suma del capital humano, el capital estructural y el capital relacional. Gestionarlo, para crear valor en la empresa, no está determinado por explotar uno de estos factores, sino de establecer una interacción entre ellos, de modo que el resultado generado sea el más adecuado para la organización.

Por consiguiente, el capital humano constituye un activo intangible que tiene la capacidad de apoyar el incremento de la productividad, impulsar la innovación y con ello la competitividad. De manera, que los recursos humanos son fuente de ventaja competitiva para las organizaciones, por ello, las empresas cada vez en mayor medida deben identificar el valor que la persona tiene para la organización y tratar de potenciarlo, al igual deben incrementar el valor que la empresa tiene para el empleado, con el fin de captar a las mejores personas y reducir el riesgo de transferibilidad de un activo tan valioso como es el capital humano [35].

La expresión capital humano y recurso humano son conceptos diferentes, [19], hacen una profunda reflexión, al respecto, las empresas deben superar la idea de recursos humanos para pasar al concepto de *capital humano*. La palabra “recurso” implica la idea de algo disponible a lo que se puede recurrir cuando sea necesario. Es algo perecedero, que se consume, se gasta, [21] comenta que los Recursos Humanos hacen referencia al conjunto de trabajadores, empleados o personal que conforma un negocio o empresa, y la administración de los Recursos Humanos, hace referencia al manejo, gestión o dirección del personal del negocio.

En cambio la palabra capital se refiere a la persona como un bien valioso, algo que pierde o gana valor en función de lo que se invierta en ella. Otro autor [20], define la teoría del capital humano “como el conjunto de las capacidades que un individuo adquiere por acumulación de conocimientos generales o específicos”. Al respecto [18], señala que el capital humano es definido como el conocimiento que los individuos adquieren durante su vida y que usan para producir buenos servicios o ideas en el mercado o fuera de él, además agrega que es un conjunto de capacidades intelectivas y motoras, hábitos, habilidades, que el hombre adquiere y se encuentran vinculados con la productividad.

Por otro lado [21] agrega que la Gestión de los Recursos Humanos, se encuentra enmarcada en un esfuerzo colectivo a los fines de lograr agenciar objetivos estratégicos tales como: confianza, compromiso, creatividad, inventiva y solidaridad

en pocas palabras en intangibles humanos. El desarrollo de este enfoque traslada a los gerentes que direccionan las personas en las organizaciones los factores claves y estratégicos para lograr los objetivos y metas propuestas, por ello, se hace necesario considerar las habilidades, destrezas y conocimientos de los Recursos Humanos que se agencian hoy en las organizaciones del siglo XXI para que las mismas sean competitivas.

Se hace necesario que dentro de la gestión de recursos humanos, se debe tomar en cuenta las funciones principales de la gestión **[17]**.

- 1.- Planificación, reclutamiento y selección del personal.
- 2.- Desarrollo de recursos humanos.
- 3.- Remuneración e incentivos.
- 4.- Seguridad e higiene.
- 5.- Relaciones con los empleados y relaciones laborales.
- 6.- Investigaciones de recursos humanos.

Se puede decir, que la Gestión de los Recursos Humanos es un Proceso sistemático, organizado, oportuno, complejo y racional para proyectar o prever los escenarios futuros y conocer o determinar, mediante la puesta en práctica de un conjunto de técnicas, la cantidad de personal necesario y competencias requeridas, así como los medios para satisfacerla con los objetivos de optimizar los beneficios y asegurar el desarrollo y la motivación individual de acuerdo con la estrategia empresarial y las exigencias y particularidades de cada cargo **[20]**.

Destaca **[21]**, que son tres los aspectos que fundamentan la Gestión de los Recursos Humanos en las organizaciones:

- *Son seres humanos*, los cuales están dotados de personalidad propia profundamente diferentes entre sí, tienen historias distintas y poseen

conocimientos, habilidades, destrezas y capacidades indispensables para la gestión adecuada de los recursos organizacionales.

- *Activadores inteligentes de los recursos organizacionales*: son elementos impulsores de la organización, capaces de dotarla de inteligencia, talento y aprendizajes indispensables en su constante renovación y competitividad en un mundo de cambios y desafíos.
- *Socios de la organización*: son capaces de conducirla a la excelencia y al éxito.

El tercer milenio, apunta a cambios cada vez más acelerados e intensos en las organizaciones y en las personas. Se destaca, que se está viviendo dentro de la sociedad del conocimiento, por lo que, es necesaria una respuesta a las condiciones de las organizaciones de un nuevo marco de actuación que proviene del desarrollo tecnológico sustentado básicamente por el aprendizaje, la tecnología de la comunicación, la información y el conocimiento, siendo el gran protagonista, el último, en un paso más para alcanzar la comprensión de la variable “humano” en el desarrollo de las organizaciones y mirando al futuro, es necesario, dar un salto cualitativo en la relación de la organización con su gente, a través de la integración de modernas tecnologías blandas, en un proceso que incorpora trabajar y atender todas las dimensiones de las personas que conforman y forman la organización [36].

5.1.2.- Administración de Recursos Humanos (ARH) y los sistemas de información

Analizando el concepto Capital Humanos se formular la siguiente pregunta ¿De qué sirven las empresas y la maquinaria, sin dirigentes ni operadores?. La tecnología puede importarse, pero los operarios y los ingenieros deben cultivarse y formarse. Una mano de obra cualificada desempeña un papel fundamental en el crecimiento económico y la productividad. Ni siquiera las que se consideran empresas de excelencia en el mundo, con gran capital humano, pueden asegurar la continuidad

de su competitividad, si no empieza a atender las necesidades sociales y psicológicas de los trabajadores [22].

Una organización que se mueve en ambientes competitivos y cuenta con capital humanos con el conocimiento experto, las competencias, las habilidades y la experiencia indispensables para competir ventajosamente en un ambiente global significarán la diferencia entre el éxito y el fracaso [22].

En conclusión, si se desea una administración de recursos humanos eficiente y eficaz. Se debe contar con capital humano y un sistema de información para la administración de la empresa o institución, que se ajuste a las necesidades de la organización.

5.1.3.- Sistemas de información para la administración de recursos humanos

Todos los cambios que se susciten alrededor en esta época contemporánea [27], explica en términos de una transición de una era a otra, la cual involucra dejar una era cultural y tecnológica y entrar en otra. Se viven las primeras etapas de un cambio en la concepción del mundo y la manera en que pensamos al respecto, así como un cambio científico y tecnológico con el que tratamos de alcanzar nuestros propósitos. La época que estamos dejando se le llama la era de las máquinas y la que estamos iniciando es la era de los sistemas.

A pesar de que las eras no tienen un principio o fin preciso, se puede decir que en los años 40's comienza el fin de la era de las máquinas y el principio de la era de los sistemas [27].

Los sistemas de información surgen aun cuando el hombre no tenía conocimientos de que los estaba utilizando para su comunicación y para el aprendizaje. Mientras recolectaba objetos sin conocer exactamente su fin, dibujaba en paredes animales, símbolos o adquiría conocimiento de los diferentes animales.

La empresa en cada una de las áreas de funcionamiento se ve en la necesidad de tomar decisiones, desde la más insignificante hasta aquella que tiene que ver con su existencia misma. Con tal responsabilidad se necesita una base de datos lo más certera posible sobre la cual descansen esas decisiones. Como es de suponer, esta información necesaria para tomar las decisiones no aparece casualmente, sino que es recopilada mediante un sistema informativo que abarca su recepción, elaboración, registro y transmisión.

Para entender lo que implica un sistema de información para la administración debemos en primer lugar establecer la diferencia entre información y dato [28].

Los datos son caracteres en forma codificada, aceptable para ser introducida y procesada. El dato es la personificación material de la información, constituye la base; es importante destacar que los datos por si solos carecen de significado, solo adquieren este cuando son interpretados convirtiéndose así en información útil [28].

La Información es el resultado del procesamiento y análisis de los datos, para que la información sea útil, ha de reunir las siguientes cualidades:

Tabla 1.- Cualidades de la información

Exactitud	Debe ser verdadera y correcta, y describir con fidelidad el objeto o hecho.
Oportunidad	Debe estar disponible cuando se necesite sin demasiado retraso.
Pertinencia	Debe relacionarse con la situación en cuestión.
Integridad	Proporcionar al investigador todos los detalles que necesite para entender la situación.
Frecuencia	Se prepara y suministra a los investigadores con bastante frecuencia para actualizarlos.
Horizontes de Tiempo	Se orienta a actividades y hechos pasados, presentes y futuros.
Alcance	Abarca en forma amplia o reducida un área de interés.
Origen	Puede generarse tanto en fuentes internas como externas.
Forma de Presentación	Las tablas de números o las presentaciones gráficas de información, son las firmas escritas o impresas más comunes. Puede incluir además una presentación verbal.

Fuente: Elaboración propia con información de [28].

Con la información adecuada una empresa es capaz de lograr un alto rendimiento e inclusive superar sus propias expectativas, para ello debe [29]:

1. Explotar los detalles mínimos: las empresas logran diferenciarse de la competencia analizando las preferencias del cliente.
2. Generalizando ideas: La capacidad de generar ideas multiplicada por la habilidad de generalizarlas representara un crecimiento para la empresa. Las ideas generales evitan la duplicidad de información y reducen el margen de error.
3. Crear un negocio de la información, en el que el producto final pasa a segundo término.
4. Una de las principales funciones de la información es proveer control a los mandos medios.
5. Evitar la información sobrante; a veces hay tanta información que es difícil separar lo relevante de lo que no lo es.
6. Contar con suficiente información; aunque exista mucha información es probable que no se cuente con la información necesaria para un proyecto específico.
7. Conocer la información que debe ser de dominio público y la que es estrictamente confidencial.
8. Conseguir que la información sea completa y exacta.
9. Lograr que la información sea oportuna.
10. Debe estar al alcance de la persona indicada.

Un *sistema* es "un conjunto de partes y elementos interrelacionados entre sí, con una meta u objetivo específico"[25], También [26] define que es un conjunto de unidades recíprocamente relacionadas. Los parámetros de los sistemas son [30]:

Tabla 2.- Parámetros de los sistemas

Parámetro	Descripción
Entrada o insumo (input)	Es la fuerza de arranque del sistema, que provee el material o la energía para la operación del sistema
Salida o resultado (output)	Es la finalidad para la cual se reunieron elementos y relaciones del sistema. Los resultados de un proceso son las salidas, las cuales deben ser coherentes con el objetivo del sistema
Procesamiento o procesador (throughput)	Es el fenómeno que produce cambios, es el mecanismo de conversión de las entradas en salidas o resultados. Generalmente es representado como la caja negra, en la que entra insumos y salen cosas diferentes, que son los productos.
Retroalimentación o retroinformación (feedback)	Es la función de retorno del sistema que tiende a comparar la salida con un criterio preestablecido, manteniéndola controlada dentro de aquel estándar o criterio
Ambiente	Es el medio que envuelve externamente el sistema. Está en constante interacción con el sistema, ya que éste recibe entradas, las procesa y efectúa salidas.

Fuente: Elaboración propia con información de [30].

Las Funciones del Sistema de Información (SI)

Para [28] el SI lleva a cabo una serie de funciones que se pueden agrupar en cuatro grupos:

1. Funciones de captación y recolección de datos. Se recopila la información externa e interna de forma continua y supervisada.
2. Almacenamiento de la información. La información relevante será almacenada para que sea accesible a quien la necesite, medidas de seguridad para evitar accesos no deseados.
3. Tratamiento de la información. Tiene por objetivo transformar la información en información significativa, para ofrecerla en el formato que sea requerida.
4. Distribución de la Información. Cada usuario debe poseer la información en el momento preciso.

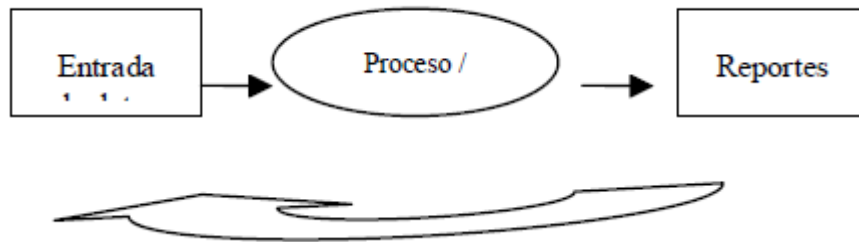


Figura 1 Actividades de un sistema de información Fuente: [28].

Por otro lado [28] menciona que un SI está compuesto por:

- ❖ Herramientas tecnológicas
- ❖ Equipo computacional
- ❖ Recurso humano.

Un buen sistema de información para la administración no reemplaza al administrador ni realiza ninguna función para la toma de decisiones [27].

Los emprendedores deben identificar que los sistemas de información apoyaran en el funcionamiento de los negocios, proporcionarán información oportuna para la toma de decisiones y deberán generar valor para el cumplimiento la visión y objetivos estratégicos [23]

El objetivo básico de la información es apoyar a la toma de decisiones dentro de la organización. Se busca tener más bases sustentables para poder decidir qué es lo que se va a hacer y qué rumbo tomar para lograr los objetivos que se planearon. Entonces los sistemas de información son una herramienta necesaria e importante para el flujo de la información en una empresa, son una fuente importante de apoyo para llevar a cabo la dirección de la misma y proporciona los conocimientos básicos para la realización de las actividades de cualquier área [23], incluso la de recursos humanos.

Los sistemas deben proporcionar información clave para la toma de decisiones; esta información debe ser sencilla, clara, expedita, veraz, precisa, consistente y fácil de analizar e interpretar [24].

Una empresa crece cuando se usan los sistemas de cómputo e internet en la operatividad de procesos y relaciones con clientes y proveedores, de otra manera administrar el negocio se vuelve complejo.

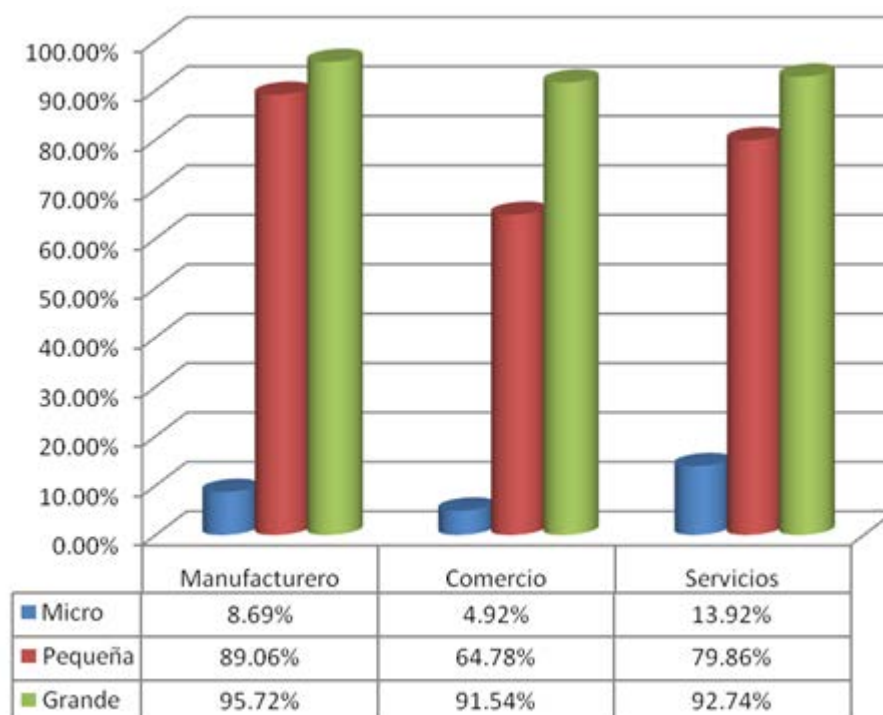


Figura 2 Porcentaje de empresas que emplean equipos de cómputo en procesos administrativos. Fuente [23].

Las estadísticas anteriores resaltan la importancia de implementar sistemas de cómputo e internet en los negocios en México, así mismo, los emprendedores deben visualizar que a medida que un negocio crece requerirá de un mayor uso de sistemas de información. Dependiendo del sector en el que operarán, deben tener claro el uso que la industria le da a la tecnología [23].

Una de las herramientas con mayor potencial para lograr obtener fuertes bases en las empresas es el uso adecuado de la información. Es por esto que el uso de

sistemas de información aporta un apoyo significativo para todas aquellas empresas que buscan un futuro exitoso [23].

Un sistema de información y control administrativo bien diseñado, debe contemplar como mínimo lo siguiente [24]:

- Ser fácil de utilizarse.
- Estar estructurado lógicamente.
- Tener procedimientos y una estructura de reportes.
- Proporcionar información completa sobre las diferentes áreas y productos de la empresa.
- Tener un mecanismo de retroalimentación que permita conocer los avances en los programas de todas las áreas.
- Asegurar que las políticas y reglas establecidas sean cumplidas.

El montaje de un sistema de información para administrar personas, requiere análisis y evaluación de la organización o de sus subsistemas y de sus respectivas necesidades de información.

El punto de partida de un sistema de información de para administrar personas es la base de datos. El objetivo final de un sistema de información para administrar personas, es suministrar a las jefaturas información acerca del personal.

Las diversas bases de datos para administrar personas conectadas entre sí permiten obtener y almacenar datos de distintos estratos o niveles de complejidad [24], deben tener las siguientes características:

1. Datos personales de cada empleado, que forman un registro de personal.
2. Datos sobre los ocupantes de cada cargo, que forman un registro de cargos.
3. Datos acerca de los empleados de cada sección, departamento o división,

que forman un registro de los mismos.

4. Datos sobre los salarios e incentivos salariales, que forman un registro de remuneración.
5. Datos acerca de los beneficios y servicios sociales, que forman un registro de beneficios.
6. Datos sobre los candidatos (registro de candidatos), sobre cursos y actividades de entrenamiento (registro de entrenamiento).

El procesamiento de datos es la actividad de acumular, agrupar y mezclar datos para transformarlos en información u obtener otra información, o la misma información bajo otra forma, para alcanzar alguna finalidad u objetivo; de manera que el procesamiento de datos puede ser [24]:

1. Manual: se efectúa manualmente con fichas, talonarios, tarjetas, entre otros, con ayuda de máquina de escribir o de calcular.
2. Semi-automático: se utilizan máquinas de contabilidad. El operador introduce fichas o tarjetas y después la máquina realiza numerosas operaciones consecutivas ya programadas sin la intervención del operador.
3. Automático: en general es realizado mediante computadoras.

El sistema de procesamiento de datos requiere de entradas (datos) para suministrar salidas (información). El procesamiento de datos en sí incluye clasificación, almacenamiento, recuperación y tratamiento de los datos. Así como, la información consiguiente para ponerla a disposición de quienes la necesitan y requieren en el momento oportuno (diaria, semanal, mensual, trimestral o anualmente), o sea en tiempo real [21].

Definición de la Administración de la Información: Disciplina encargada de analizar la información como un recurso en las organizaciones. Cubre la definición, uso, valor y distribución de todos los datos en la organización independientemente si

esta es procesada por sistemas computacionales o no. Evalúa el tipo de datos/información que la organización requiere para funcionar eficientemente **[28]**.

Su importancia radica en los siguientes puntos **[28]**:

1. Incremento “exponencial” en la cantidad de información administrada por las organizaciones.
2. Reducción en precio en los equipos de cómputo.
3. Proliferación de sistemas desde hojas de cálculo hasta sistemas empresariales.
4. Mayor Competencia.

La Administración de la Información (AI) por lo tanto tiende a involucrar **[28]**:

1. Tecnología
2. Procesos
3. Personal
4. Contenido

Cada uno de estos elementos deben de ser considerados si se pretende tener éxito en esta área.

Alguno de los problemas típicos relacionados con la AI son **[28]**:

1. Poca coordinación e integración entre sistemas.
2. Falta de visión estratégica en el área tecnológica.
3. Adopción limitada de los sistemas de cómputo en la organización.
4. Falta de calidad en la información, duplicación, inconsistencia, no actualizada, entre otros.
5. Falta de recursos.
6. Falta de apoyo y desconfianza en los usuarios.

5.1.4.- Influencia de la Tecnología en la Administración de Recursos Humanos (ARH).

La tecnología de la información ha modificado el aspecto de la ARH. Tal vez la utilización más importante de la tecnología en ARH sea un sistema de información de capital humano (SIRH) de la organización. Como puede verse en la definición (método computarizado que proporciona información correcta y actualizada de los empleados para fines de control y toma de decisiones), rebasa el simple almacenamiento y recuperación de información para incluir aplicaciones más amplias como producción de reportes, previsión de las necesidades de RH, planeación estratégica, planeación y *ascenso dentro de la organización* y plan de carrera y evaluación de las políticas y prácticas de RH [21].

Una empresa que quiera competir en el mercado donde se desarrolla necesita contar con sistemas de información que le provean la información necesaria para realizar sus operaciones en una forma rápida y eficiente, no basta con tener la información, hay que manejarla a favor de la empresa eficientemente si es que se quiere tener competitividad, las empresas con gran cantidad de utilidades han sido aquellas que cuentan con una adecuada organización y que cuentan con sistemas de información de calidad, esto no significa que deba ser desarrollado por grandes empresas o de alto costo, ya que hasta los grandes desarrolladores se equivocan [28].



Las empresas con sistemas de información de calidad son aquellas que tiene una cultura organizacional flexible, es decir, hacen cambios en su forma de hacer las cosas, hacen uso de la tecnología para el bien de la empresa [28].


Dentro de unos pocos años aquellas empresas que no cambien sus viejos sistemas o no sean capaces de desarrollar dentro de ellas sistemas de información de calidad, o no hagan uso adecuado de la información y tengan una apertura hacia


nuevas ideas y uso de nuevas tecnologías simplemente serán aplastadas por la ignorancia al cambio [24].

Hay que planear el crecimiento del sistema y la compra del equipo que se va a necesitar para que todos los equipos que se adquieran sean compatibles. Cuando un sistema de información y de control no es diseñado profesionalmente y por gente que ya conoce la empresa, puede resultar poco práctico y no responder a las necesidades para las que fue adquirido. En la actualidad, existen programas computacionales con los que se puede manejar la información administrativa [24]. Algunos de los sistemas se muestran en la Tabla 3.

Tabla 3.- Sistemas de información de recursos humanos en el mercado

Nombre	Beneficios	Lenguaje	Costo
	<p>Ayuda a mejorar la administración de su capital humano.</p> <p>Permite al personal de recursos humanos el acceso a la información de todos los empleados de manera rápida y sencilla.</p> <p>Ayuda en la evaluación y seguimiento de empleados prospectos.</p> <p>Permite visualizar rápidamente el historial de salarios y el historial académico y laboral de cada empleado, facilitando la evaluación y revisión de salarios.</p>	<p>Compatible con servidores Windows® y Linux®.</p> <p>Software distribuido bajo una licencia <u>Open Source</u></p> <p>Aplicación Web Dinámica Java™.</p> <p>Arquitectura Modelo, Vista, Controlador (MVC)</p>	\$5,500.00
	<p>Permite realizar un mantenimiento de la aplicación directamente por el usuario</p>	<p>Bajo la tecnología Be Microsoft.NET</p>	\$6,600.00

	<p>Mejorar el aprovechamiento del recurso humano</p> <p>Contar con información oportuna</p> <p>Eliminar papeleo</p> <p>Reducir costos de operación</p> <p>Reducción de esfuerzo y eliminar trabajos manuales</p> <p>Agiliza las operaciones</p> <p>Asegura la confidencialidad de la información</p> <p>Asegura la confiabilidad de los datos.</p> <p>Sistema integral conformado por diferentes módulos, en donde su cobertura y alcance engloba las diferentes áreas o disciplinas de la administración de los recursos humanos en las organizaciones.</p> <ul style="list-style-type: none"> • Fortia Nómina • Fortia Reloj (Tiempo y Asistencia) • Fortia Recursos Humanos • Fortia Kiosco • Fortia Designer (Reporteador) • Fortia Indiktors (Tablero de mandos) 		
	<p>Fácil de usar, su interface fácil e intuitiva traducida en pantallas y menús fáciles de usar agiliza y ayuda a que tus procesos de administración</p>	<p>Servicios en la Nube (SaaS)</p>	<p>¡Paga solo por lo que usa! El gasto en base al</p>

	de Recursos Humanos, cálculo de nómina y control de asistencia sean más sencillos que nunca. Cuenta con las mejores prácticas de negocio en la automatización e implementación de estas soluciones.		número de empleados que maneje
	Permite visualizar todas las requisiciones de personal pendientes de cubrir. La administración del personal ocupa el lugar central de la aplicación y es aquí en donde se efectúa la actualización de los colaboradores; sus datos personales, laborales y sus cambios en la organización. El usuario de cada módulo puede crear y diseñar reportes a la medida de sus necesidades	Diseñado para Microsoft Windows Base de Datos MySQL Visual studio 2005, C#	\$5,700.00

Fuente: Elaboración propia con datos de [31], [32], [33], [34]

Ahora bien, la selección de los programas o la creación de uno específico para la empresa, está determinado por el giro y las necesidades de la misma [24].

El manejo cotidiano de dichos sistemas permite definir las necesidades de control de cada empresa y adaptar los reportes al sistema de información con que se cuenta, a fin de que en él quede incluida la mejor información posible para que las personas indicadas tomen las decisiones correctas [24].

Diseñar todo tipo de sistemas y procedimientos que controlen la información y los movimientos producidos, es fundamental para pensar en el éxito de una empresa [24].

La información es un valioso “poder” que determina el éxito o fracaso de las empresas, y por lo tanto de los que en ella colaboran, sin embargo este éxito puede ser no favorable, dependiendo de los conocimientos y principalmente de los valores de cada individuo. A pesar de los grandes desarrollos tecnológicos de la actualidad, de los avances en el área computacional y de la creación de herramientas que hace pocos años eran impensables el tratamiento de la información y su control empieza dentro del recurso humano y es éste el que determinará el rumbo final de cada empresa [21].

5.2.- Desarrollo de un sistema de información.

Todo sistema organizacional depende, en mayor o menor medida, de una entidad abstracta denominada sistema de información. Este sistema es el medio por el cual los datos fluyen de una persona o departamento hacia otros y pueden ser cualquier cosa, desde la comunicación interna entre los diferentes componentes de la organización y líneas telefónicas hasta sistemas de cómputo que generan reportes periódicos para varios usuarios [8].

Los sistemas de información están formados por subsistemas que incluyen hardware, software, medios de almacenamiento de datos para archivos y bases de datos. El conjunto particular de subsistemas utilizados – equipo específico, programas, archivos y procedimientos- es lo que se denomina una aplicación de sistemas de información [8].

Un sistema de información acepta datos provenientes del entorno, los procesa y genera datos de salida para la toma de decisiones.

5.2.1.- Características de un sistema de información

De acuerdo a [7], las características de un sistema de información, puede agruparse en:

- A) Tecnologías, son las que afectan al rendimiento y seguridad del sistema desde el punto de vista del equipo.
- B) Funcionales y semánticas, se refiere a que el sistema realiza y funcione de forma correcta (eficacia) y si es capaz de adaptarse a requisitos cambiantes.
- C) Económicas, esta característica da énfasis al costo del sistema y a la eficacia con que responden a todos los componentes del sistema.
- D) Sociales, son las que tienen un impacto sobre el entorno social (interno o externo) en que se desenvuelve el sistema.

5.2.2.- Categorías de sistemas de información

Según [8] y [10], el análisis de sistemas, arroja diferentes tipos de sistemas de información para satisfacer las diversas necesidades de una empresa y las clasifican en tres grandes rubros, los cuales son:

Sistema para el procedimiento de transacciones (TPS) Tiene como finalidad mejorar las actividades rutinarias de una empresa y de las que dependen toda la organización. Una transacción es cualquier suceso o actividad que afecta a toda la organización. Las transacciones más comunes incluyen facturación, entrega de mercancía, pago a empleados y depósito de cheques.

Los sistemas de procesamiento de transacciones brindan velocidad y exactitud; además se pueden programar para seguir rutinas sin ninguna variación [10].

Sistemas de información administrativa (MIS) Ayuda a los directivos a tomar decisiones y resolver problemas. Los directivos recurren a los datos almacenados como consecuencia del procesamiento de las transacciones. En cualquier organización se deben tomar decisiones sobre muchos asuntos que se presentan con regularidad (a la semana, al mes, al trimestre, etc.) y para hacerlo se requiere de cierta información. Dado que los procesos de decisión están claramente definidos, entonces se pueden identificar la información necesaria para formular las

decisiones. Se pueden desarrollar sistemas de información para que, en forma periódica, preparen reportes para el soporte de decisiones. Cada vez que se necesita la información, está se prepara y presenta en una forma y formato diseñados con anterioridad [8].

Sistema para el soporte de decisiones (DSS) Ayudan a los directivos que deben tomar decisiones no muy estructuradas, también denominadas no estructuradas o decisiones semiestructuradas. Una decisión se considera no estructurada si no existen procedimientos claros para tomarla y tampoco es posible identificar, con anticipación, todos los factores que deben considerarse en la decisión [8].

El sistema de información que se presenta en este trabajo escrito se refiere a un Sistema de Información Administrativo (MIS), porque proporciona información y reportes de los empleados que trabajan en el SAT, donde se puede consultar los movimientos de altas y bajas, cambios de nivel y categorías, entre otros, para la toma de decisiones por parte del administrador quien se encarga de la contratación del personal en el SAT.

5.2.3.- Estrategias para el desarrollo de sistemas

Los sistemas de información basados en computadoras sirven para diversas finalidades que van desde el procedimiento de las transacciones de una empresa, hasta proveer de la información necesaria para definir sobre asuntos que se presentan con frecuencia.

Todas estas situaciones están representadas por tres distintos enfoques al desarrollo de sistema de información basados en computadora [9], [11] y [8]:

- 1.- Método del ciclo de vida para el desarrollo de sistemas
- 2.- Método de desarrollo del análisis estructurado
- 3.- Método del prototipo de sistemas

Estos se describirán a detalle en el apartado de procesos de software.

5.2.4.- Análisis de Requerimientos

Una introducción a la identificación de requerimientos

A través de los años se ha podido constatar que los requerimientos o requisitos son la pieza fundamental en un proyecto de desarrollo de software, ya que marcan el punto de partida para actividades como la planeación, básicamente en lo que se refiere a las estimaciones de tiempos y costos, así como la definición de recursos necesarios y la elaboración de cronogramas que será uno de los principales mecanismos de control con los que se contará durante la etapa de desarrollo [37].

[15] Menciona que para desarrollar un sistema, se debe identificar correctamente, analizar y entender cuáles son los requerimientos de los usuarios o lo que los usuarios quieren que haga el sistema. El proceso y las técnicas que un analista de sistemas usa para identificar, analizar y entender requerimientos de sistema, son referidos como la identificación de requerimientos.

Gran número de los proyectos de software fracasan por no realizar una adecuada definición, especificación, y administración de los requerimientos. Dentro de esa mala administración se pueden encontrar factores como la falta de participación del usuario, requerimientos incompletos y el mal manejo del cambio a los requerimientos [37].

5.2.5.- Requerimientos del sistema

[37], [12] y el glosario de la IEEE, definen qué es un “Requerimiento”, como sigue:

1. “Una condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo”.

2. “Una condición o capacidad que debe estar presente en un sistema o componentes de sistema para satisfacer un contrato, estándar, especificación u otro documento formal”.
3. “Un requerimiento es simplemente una declaración abstracta de alto nivel de un servicio que debe proporcionar el sistema o una restricción de éste” [37].
4. “Los requerimientos del sistema especifican lo que el sistema de información deberá hacer o cual propiedad o cualidad debe de tener este” [12].

De acuerdo a [37] un requerimiento es una descripción de una condición o capacidad que debe cumplir un sistema, ya sea derivada de una necesidad de usuario identificada, o bien, estipulada en un contrato, estándar, especificación u otro documento formalmente impuesto al inicio del proceso.

Según [38] el objetivo principal de los requerimientos es la comprensión de lo que los clientes y los usuarios esperan que haga el sistema, el cual expresa el propósito del sistema sin considerar como se va a implantar. En otras palabras, los requerimientos identifican el qué del sistema, mientras que el diseño establece el cómo del sistema.

La parte más difícil de construir un sistema es precisamente saber qué construir. Ninguna otra parte del trabajo conceptual es tan difícil como establecer los requerimientos técnicos detallados, incluyendo todas las interfaces con gente, máquinas, y otros sistemas. Ninguna otra parte del trabajo afecta tanto al sistema si no se definen correctamente los requerimientos del mismo. Ninguna es tan difícil de corregir más adelante. Por ello, la tarea más importante que el ingeniero de software hace para el cliente es la extracción iterativa y el refinamiento de los requerimientos del producto [39].

Esencialmente, el protocolo de la identificación de los requerimientos y su administración es identificar correctamente los requerimientos de conocimiento, proceso y comunicación para los usuarios de un sistema nuevo. La problemática de

identificar correctamente los requerimientos del sistema puede dar como resultado una o más de las siguientes fallas **[15]**:

- El costo del sistema puede aumentar más de lo proyectado.
- El sistema puede ser entregado después de lo prometido.
- El sistema puede no estar a la altura de las expectativas de los usuarios y esa insatisfacción puede originar que no lo usen.
- Una vez en producción, los costos de mantenimiento y mejora del sistema pueden ser excesivamente altos.
- El sistema puede ser poco confiable y tener la tendencia a fallar y tener mucho tiempo muerto.

El análisis de requerimientos es el conjunto de técnicas y procedimientos que permiten conocer los elementos necesarios para definir un proyecto de software. Es una tarea de ingeniería del software que permite especificar las características operacionales del software, indicar la interfaz del software con otros elementos del sistema y establecer las restricciones que debe cumplir el software **[38]**.

La tarea de análisis de los requerimientos es un proceso de descubrimiento y refinamiento, el cliente y el desarrollador tienen un papel activo en la ingeniería de requerimientos de software. El cliente intenta plantear un sistema que en muchas ocasiones es confuso para él, sin embargo, es necesario que describa los datos, que especifique las funciones y el comportamiento del sistema que desea. El objetivo es que el desarrollador actúe como un negociador, un interrogador, un consultor, o sea, como persona que consulta y propone para resolver las necesidades del cliente **[38]**.

5.2.6.- Lineamientos al definir los requerimientos

Al definir los requerimientos del sistema [14] y [15] señalan que es importante que se obedezcan los siguientes lineamientos:

Tabla 4.- Lineamientos de los requerimientos

Consistentes:	Los requerimientos en el documento no deben contradecirse.
Complejos:	Los requerimientos describen todas las posibles entradas del sistema y las respuestas.
Factibles:	Los requerimientos pueden satisfacer con los recursos disponibles y sus restricciones.
Requeridos:	Los requerimientos se necesitan en realidad y cumplen con el objetivo del sistema.
Exactos:	Los requerimientos son expresados de manera correcta.
Rastreables:	Los requerimientos apuntan directamente hacia las funciones y características del sistema.
Verificables:	Los requerimientos se definen de formar en que pueden ser comprobados durante la prueba.
Realismo	Utilizando el conocimiento en la tecnología los requerimientos deben verificarse para asegurar que se puedan implementar. Estas verificaciones también deben tener en cuenta el presupuesto y la confección de agendas para el desarrollo del sistema.

Fuente: Elaboración propia con datos de [14] y [15].

Este proceso puede ser difícil, frustrante y tomar mucho tiempo; por ello a menudo se toman atajos para ahorrar tiempo y dinero. Sin embargo este criterio tan estrecho a menudo conduce a los problemas mencionados anteriormente. Ahora bien analizaremos el proceso.

5.2.7.- Características de los requerimientos

Los requerimientos permiten que los desarrolladores expliquen cómo han entendido lo que el cliente pretende del sistema. También, indican a los diseñadores qué funcionalidad y que características va a tener el sistema resultante. Y además, indican al equipo de pruebas qué demostraciones llevar a cabo para convencer al

cliente de que el sistema que se le entrega es lo que solicitó. Las características de los requerimientos mencionados en el estándar IEEE830 [38] lo explican como sigue:

Tabla 5.- Características de los requerimientos

Deben ser correctos	Tanto el cliente como el desarrollador deben revisarlos para asegurar que no tienen errores.
Debe ser conciso	Un requerimiento es conciso si es fácil de leer y entender. Su redacción debe ser simple y clara para aquellos que vayan a consultarlo en un futuro.
Deben ser consistentes	Dos requerimientos son inconsistentes cuando es imposible satisfacerlos simultáneamente.
Deben estar completos	El conjunto de requerimientos está completo si todos los estados posibles, cambios de estado, entradas, productos y restricciones están descritos en alguno de los requerimientos, es decir si se proporcionó la información suficiente para su comprensión.
Deben ser realistas.	Todos los requerimientos deben ser revisados para asegurar que son posibles.
No ambiguo	Un requerimiento no es ambiguo cuando tiene una sola interpretación. El lenguaje usado en su definición, no debe causar confusiones al lector.
¿Cada requerimiento describe algo que es necesario para el cliente?	Los requerimientos deben ser revisados para conservar sólo aquellos que inciden directamente en la resolución del problema del cliente.
Deben ser verificables	Se deben poder preparar pruebas que demuestren que se han cumplido los requerimientos.
Deben ser rastreables	¿Se puede rastrear cada función del sistema hasta el conjunto de requerimientos que la establece?

Fuente: Elaboración propia con datos de [38] y [37].

5.2.8.- Dificultades para definir los requerimientos

Durante la etapa de especificación de requerimientos se pueden presentar muchos inconvenientes los cuales son importantes de identificar y prevenir, a continuación se presenta en la Figura 3, los problemas más comunes en este proceso:

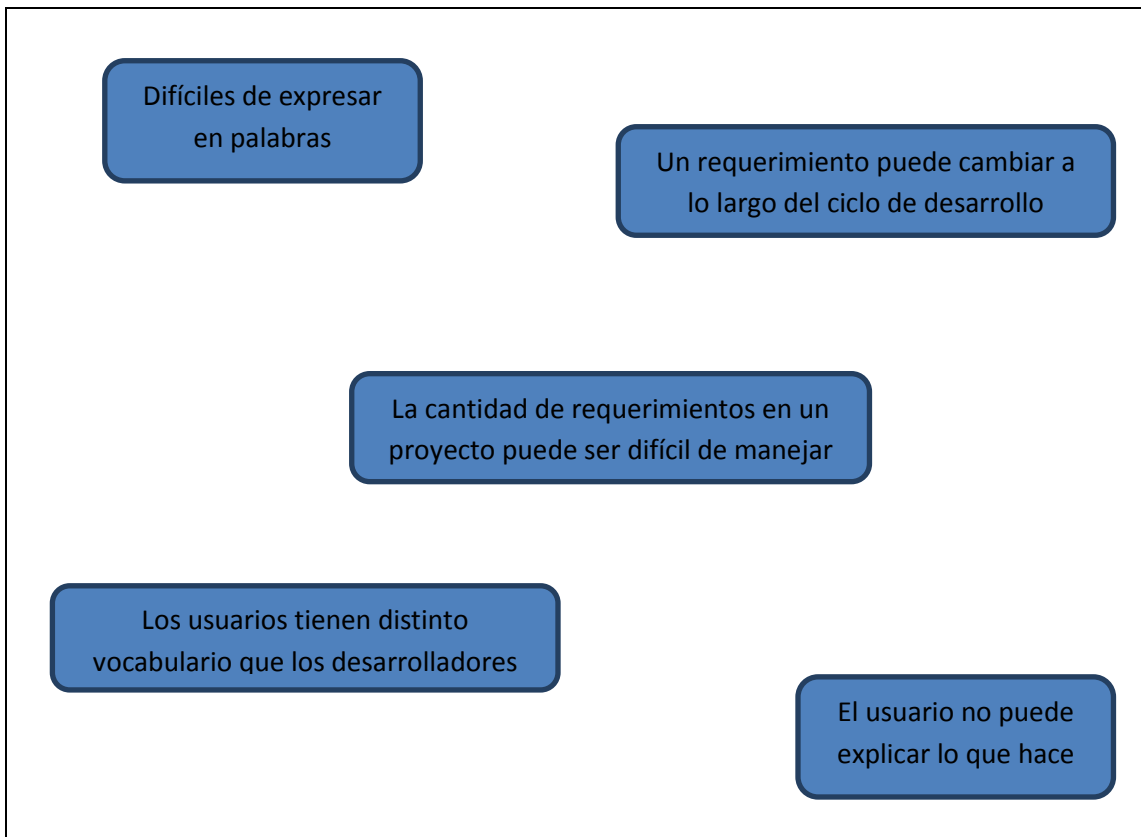


Figura 3 Dificultades de los requerimientos. Fuente: Elaboración propia con datos de [37].

La comunicación entre los usuarios y el desarrollador es fundamental para el desarrollo del sistema, pero en ocasiones pueden surgir dificultades. De tal manera que entre el desarrollador y el usuario puede existir discrepancia, por ello [38] señala: "Sé que crees que comprendes lo que piensas que he dicho, pero no estoy seguro de que entendiste lo que yo quise decir". En la Tabla 6 se muestran las relaciones entre desarrollador –usuario y usuario – desarrollador.

Tabla 6.- Comparación de relaciones Desarrollador-Usuario y Usuario-Desarrollador

Como ven los desarrolladores a los usuarios	Como ven los usuarios a los desarrolladores
Los usuarios no saben lo que quieren	Los desarrolladores no comprenden las necesidades operacionales
Los usuarios no pueden articular lo que quieren	Los desarrolladores ponen demasiado énfasis en la técnica
Los usuarios tienen muchas necesidades motivadas políticamente	Los desarrolladores pretenden decirnos como hacer nuestro trabajo
Los usuarios lo quieren todo bien y ahora	Los desarrolladores no pueden traducir nuestras necesidades claramente

	establecidas a un sistema exitoso
Los usuarios son incapaces de priorizar sus necesidades	Los desarrolladores dicen “no” todo el tiempo
Los usuarios rehúsan tomar responsabilidades por el sistema	Los desarrolladores siempre están por encima del presupuesto
Los usuarios son incapaces de proporcionar un enunciado utilizable de las necesidades	Los desarrolladores siempre están atrasados
Los usuarios no están comprometidos con los proyectos de desarrollo de sistemas	Los desarrolladores piden a los usuarios tiempo y esfuerzo, aún en detrimento de sus obligaciones primarias importantes
Los usuarios no tienen voluntad de colaborar	Los desarrolladores establecen estándares no realistas para la definición de los requerimientos
Los usuarios no pueden mantener el cronograma	Los desarrolladores son incapaces de responder rápidamente a los legítimos cambios de las necesidades

Fuente: Elaboración propia con datos de [38].

5.2.9.- Tipos de Requerimientos

Para [37] y [38] los requerimientos de software pueden dividirse en 2 categorías: requerimientos funcionales y requerimientos no funcionales.

Tabla 7.- Tipos de requerimientos

FUNCIONALES	NO FUNCIONALES
<p>Son los que definen las funciones que el sistema será capaz de realizar, describen las transformaciones que el sistema realiza sobre las entradas para producir salidas. Es importante que se describa el ¿Qué? y no el ¿Cómo? se deben hacer esas transformaciones. Estos requerimientos al tiempo que avanza el proyecto de software se convierten en los algoritmos, la lógica y gran parte del código del sistema.</p> <p><i>Los requerimientos funcionales de un sistema describen lo que el sistema debe hacer.</i></p>	<p>Describen una restricción sobre el sistema que limita nuestras elecciones en la construcción de una solución al problema. Restringen los servicios o funciones ofrecidas por el sistema. Incluyen restricciones de tiempo, el tipo de proceso de desarrollo a utilizar, fiabilidad, tiempo de respuesta, capacidad de almacenamiento.</p> <p><i>Los requerimientos no funcionales ponen límites y restricciones al sistema.</i></p>

Fuente: Elaboración propia con datos de [37] y [38].

5.2.10.- Técnicas y herramientas utilizadas en la Ingeniería de Requerimientos (IR)

Existen varias técnicas para la IR, [37] abarca cinco de ellas. Es importante resaltar que estas técnicas pueden ser aplicables a las distintas fases del proceso de la IR, haciendo la salvedad de que hay que tomar en cuenta las características propias del proyecto en particular que se esté desarrollando para aprovechar al máximo su utilidad.

- 1) Entrevistas y Cuestionarios
- 2) Sistemas existentes
- 3) Lluvia de ideas (Brainstorm)
- 4) Prototipos
- 5) Casos de Uso

Las entrevistas sirven para obtener una comprensión general de lo que hacen los futuros usuarios del sistema, cómo podrían interactuar con el sistema y las dificultades a las que se enfrentan con los sistemas actuales. El cuestionario permite que el entrevistado conozca los temas que se van a tratar y pueda conseguir con anticipación información que no tenga a disposición inmediata [38].

Tipos de entrevistas

1. *Entrevista cerrada: las preguntas ya están previstas, tienen un orden y una forma de ser planteadas que no pueden ser modificadas por el entrevistador. Es en realidad un cuestionario.*
2. *Entrevista abierta: el entrevistador tiene amplia libertad para las preguntas o para las intervenciones permitiendo toda la flexibilidad necesaria para cada caso particular [13].*

Sistemas existentes consiste en analizar distintos sistemas ya desarrollados que estén relacionados con el sistema a ser construido. Por un lado, podemos analizar

las interfaces de usuario, observando el tipo de información que se maneja y cómo es manejada, por otro lado también es útil analizar las distintas salidas que los sistemas producen (listados, consultas, etc.), porque siempre pueden surgir nuevas ideas sobre la base de estas [37].

Lluvia de ideas se usa para generar ideas. La intención en su aplicación es la de generar la máxima cantidad posible de requerimientos para el sistema. No hay que detenerse en pensar si la idea es o no del todo utilizable. La intención de este ejercicio es generar, en una primera instancia, muchas ideas. Luego, se irán eliminando en base a distintos criterios las reglas básicas a seguir según [37] son:

- Los participantes deben pertenecer a distintas disciplinas y, preferentemente, deben tener mucha experiencia.
- Conviene suspender el juicio crítico y se debe permitir la evolución de cada una de las ideas, porque si no se crea un ambiente hostil que no alienta la generación de ideas.
- Por más locas o salvajes que parezcan algunas ideas, no se las debe descartar, porque luego de maduras probablemente se tornen en un requerimiento sumamente útil.
- A veces ocurre que una idea resulta en otra idea, y otras veces podemos relacionar varias ideas para generar una nueva.
- Escribir las ideas sin censura.

Prototipos para validar los requerimientos hallados, se construyen prototipos. Los prototipos son simulaciones del posible producto, que luego son utilizados por el usuario final.

El desarrollo del prototipo comienza con la captura de requerimientos. Desarrolladores y clientes se reúnen y definen los objetivos globales del software, identifican todos los requerimientos que son conocidos, y señalan áreas en las que será necesaria la profundización en las definiciones. Luego de esto, tiene lugar un

“diseño rápido”. El diseño rápido se centra en una representación de aquellos aspectos del software que serán visibles al usuario (por ejemplo, entradas y formatos de las salidas). El diseño rápido lleva a la construcción de un prototipo [37].

Casos de Uso son una técnica para especificar el comportamiento de un sistema, sirven para especificar la funcionalidad y el comportamiento de un sistema mediante su interacción con los usuarios y/o otros sistemas [37].

La mayoría de los requerimientos funcionales, sino todos, se pueden expresar con casos de uso.

5.3.- Ingeniería de Software

Software es el conjunto de programas de cómputo, documentos asociados y esquemas de configuración necesarios para que estos programas operen [53].

Para poder comprender lo que es el software (y consecuentemente la ingeniería del software), es importante examinar las características del software que lo diferencian de otras cosas que los hombres pueden construir [71].

1.- El software se desarrolla, no se fabrica en un sentido clásico

Aunque existen similitudes entre el desarrollo del software y la construcción del hardware, ambas actividades son fundamentalmente diferentes. En ambas actividades la buena calidad se adquiere mediante un buen diseño, pero la fase de construcción del hardware puede introducir problemas de calidad que no existen (o son fácilmente corregibles) en el software. Los costes del software se encuentran en la ingeniería. Esto significa que los proyectos de software no se pueden gestionar como si fueran proyectos de fabricación.

2. El software no se «estropea».

El hardware exhibe relativamente muchos fallos al principio de su vida (estos fallos son atribuibles normalmente a defectos del diseño o de la fabricación); una vez corregidos los defectos, permanece durante un cierto periodo de tiempo. Sin embargo, conforme pasa el tiempo, el hardware empieza a desgastarse y la tasa de fallos se incrementa. El software no es susceptible a los males del entorno que hacen que el hardware se estropee. Los defectos no detectados harán que falle el programa durante las primeras etapas de su vida. Sin embargo, una vez que se corrigen (suponiendo que no se introducen nuevos errores) permanece, el software no se estropea. ¡Pero se deteriora! Es decir el software sufre cambios (mantenimiento). Conforme se hacen los cambios, es bastante probable que se introduzcan nuevos defectos, se solicita otro cambio. Lentamente, el nivel mínimo de fallos comienza a crecer -el software se va deteriorando debido a los cambios.

Otro aspecto de ese deterioro ilustra la diferencia entre el hardware y el software. Cuando un componente de hardware se estropea se sustituye por una pieza de repuesto. No hay piezas de repuesto para el software. Cada fallo en el software indica un error en el diseño o en el proceso mediante el que se tradujo el diseño a código máquina ejecutable. Por tanto, el mantenimiento del software tiene una complejidad considerablemente mayor que la del mantenimiento del hardware.

3. Aunque la industria tiende a ensamblar componentes, la mayoría del software se construye a medida.

Consideremos la forma en la que se diseña y se construye el hardware de control para un producto basado en computadora. El ingeniero de diseño construye un sencillo esquema de la circuitería digital, hace algún análisis fundamental para asegurar que se consigue la función adecuada y va al armario donde se encuentran los catálogos de componentes digitales. Después de seleccionar cada componente, puede solicitarse la compra.

5.3.1.- Aplicaciones del Software

El software puede aplicarse en cualquier situación en la que se haya definido previamente un conjunto específico de pasos procedimentales (es decir, un algoritmo). El contenido y el determinismo de la información son factores importantes a considerar para determinar la naturaleza de una aplicación de software. El contenido se refiere al significado y a la forma de la información de entrada y salida [71].

El determinismo de la información se refiere a la predecibilidad del orden y del tiempo de llegada de los datos. Algunas veces es difícil establecer categorías genéricas para las aplicaciones del software que sean significativas. Conforme aumenta la complejidad del software, es más difícil establecer compartimentos nítidamente separados. Las siguientes áreas del software indican la amplitud de las aplicaciones potenciales [71]:

Tabla 8.- Aplicaciones del software

Software	Descripción
Software de sistemas	Es un conjunto de programas que han sido escritos para servir a otros programas. Algunos programas de sistemas (por ejemplo: compiladores, editores y utilidades de gestión de archivos) procesan estructuras de información complejas pero determinadas. Otras aplicaciones de sistemas (por ejemplo: ciertos componentes del sistema operativo, utilidades de manejo de periféricos, procesadores de telecomunicaciones) procesan datos en gran medida indeterminados.
Software de tiempo real	Es el software que coordina/analiza/controla sucesos del mundo real conforme ocurren. Entre los elementos del software de tiempo real se incluyen: un componente de adquisición de datos que recolecta y da formato a la información recibida del entorno externo, un componente de análisis que transforma la información según lo requiera la aplicación, un componente de control salida que responda al entorno externo, y un componente de monitorización que coordina todos los demás componentes, de forma que pueda mantenerse la repuesta en tiempo real.
Software de gestión	El proceso de la información comercial constituye la mayor de las áreas de aplicación del software. Los «sistemas» discretos (por ejemplo: nóminas, cuentas de haberes-débitos, inventarios, entre otros.) han evolucionado hacia el software de sistemas de información de gestión (SIG) que accede a

	una o más bases de datos que contienen información comercial. Las aplicaciones en esta área reestructuran los datos existentes para facilitar las operaciones comerciales o gestionar la toma de decisiones. Además de las tareas convencionales de procesamiento de datos, las aplicaciones de software de gestión también realizan cálculo interactivo (por ejemplo: el procesamiento de transacciones en puntos de ventas).
Software de ingeniería y científico	Está caracterizado por los algoritmos de «manejo de números». Las aplicaciones van desde la astronomía a la vulcanología, desde el análisis de la presión de los automotores a la dinámica orbital de las lanzaderas espaciales y desde la biología molecular a la fabricación automática. Sin embargo, las nuevas aplicaciones del área de ingeniería/ciencia se han alejado de los algoritmos convencionales numéricos. El diseño asistido por computadora (del inglés CAD), la simulación de sistemas y otras aplicaciones interactivas, han comenzado a coger características del software de tiempo real e incluso del software de sistemas.
Software empotrado o	Los productos inteligentes se han convertido en algo común en casi todos los mercados de consumo e industriales. El software empotrado reside en memoria de sólo lectura y se utiliza para controlar productos y sistemas de los mercados industriales y de consumo. El software empotrado puede ejecutar funciones muy limitadas y curiosas (por ejemplo: el control de las teclas de un horno de microondas) o suministrar una función significativa y con capacidad de control (por ejemplo: funciones digitales en un automóvil, tales como control de la gasolina, indicadores en el salpicadero, sistemas de frenado, entre otros.).
Software de PC's	El mercado del software de computadoras personales ha germinado en las pasadas dos décadas. El procesamiento de textos, las hojas de cálculo, los gráficos por computadora, multimedia, entretenimientos, gestión de bases de datos, aplicaciones financieras, de negocios y personales y redes o acceso a bases de datos externas son algunas de los cientos de aplicaciones.
Software basado en Web	Las páginas Web buscadas por un explorador son software que incorpora instrucciones ejecutables (por ejemplo, CGI, HTML, Perl, o Java), y datos (por ejemplo, hipertexto y una variedad de formatos de audio y visuales). En esencia, la red viene a ser una gran computadora que proporciona un recurso software casi ilimitado que puede ser accedido por cualquiera con un modem.
Software de Inteligencia Artificial	Hace uso de algoritmos no numéricos para resolver problemas complejos para los que no son adecuados el cálculo o el análisis directo. Los sistemas expertos, también llamados sistemas basados en el conocimiento, reconocimiento de patrones (imágenes y voz), redes neuronales artificiales, prueba de teoremas, y los juegos son representativos de las aplicaciones de esta categoría.

Fuente: Elaboración propia [53], [54].

Ahora bien [71] define un proceso de software como un marco de trabajo de las tareas que se requieren para construir software de alta calidad. ¿Es «proceso» sinónimo de ingeniería del software?.

La respuesta es «sí» y «no». Un proceso de software define el enfoque que se toma cuando el software es tratado por la ingeniería. Pero la ingeniería del software también comprende las tecnologías que tiene el proceso -métodos técnicos y herramientas automatizadas-. Aún más importante es que la ingeniería del software la realizan personas creativas, con conocimiento, que deberían trabajar dentro de un proceso del software definido y avanzado que es apropiado para los productos que construyen y para las demandas de su mercado.

[71] dice que Ingeniería del Software es el establecimiento y uso de principios robustos de la ingeniería a fin de obtener económicamente software que sea fiable y que funcione eficientemente sobre máquinas reales.

Según [53], Ingeniería del Software es la aplicación práctica del conocimiento científico en el diseño y construcción de programas de computadora y la documentación asociada requerida para desarrollar, operar y mantenerlos. Se conoce también como desarrollo de software o producción de software.

Y [54], argumenta que ingeniería del software es la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación (funcionamiento) y mantenimiento del software; es decir, la aplicación de ingeniería al software.

5.3.2.- Proceso de Software

Es un conjunto de actividades y resultados asociados, que generan un producto de software, las cuales son llevadas a cabo por los ingenieros de software [53].

Es una abstracción de un proceso real. Existe una gran variedad de modelos o paradigmas de desarrollo de software.

Sea cual fuere el proceso utilizado y aplicado al desarrollo del software, casi siempre libremente de este proceso, se debe aplicar un modelo de ciclo de vida. Según varias fuentes consultadas se estima que, del total de proyectos software grandes emprendidos, un 28% fracasan, un 46% caen en severas modificaciones que lo retrasan y un 26% son totalmente exitosos [54].

Un modelo para el desarrollo de software es una perspectiva de las actividades que ocurren durante el diseño y el desarrollo del software, se pretende determinar el orden de las etapas implicadas en el sistema y los criterios de transición asociadas entre estas etapas, un modelo de ciclo de vida del software [54]:

- Describe las etapas primordiales del desarrollo de software.
- Define las etapas primarias esperadas para ser aplicadas durante esas etapas.
- Ayuda a administrar el progreso del desarrollo.
- Provee un espacio de trabajo para la definición de un detallado proceso de desarrollo de software.

Así, los modelos por una parte proveen una guía a los ingenieros de software con el fin de establecer las diversas actividades técnicas en el proyecto, por otra parte suministran un marco para la administración del desarrollo y el mantenimiento del software, en el sentido en que permiten estimar recursos, definir puntos de control intermedios, monitorear el avance, entre otros [54].

5.3.3.- Modelos de proceso de software

[71] argumenta que la gestión eficaz de un proyecto de software se centra en las cuatro P's: "personal", "producto", "proceso" y "proyecto". El orden no es arbitrario. Tienen una influencia sustancial en la gestión de proyectos de software.

La primera P's, "personal", se debe identificar el «factor humano» para el desarrollo del software altamente preparado. El proceso del software (y todos los proyectos de software) lo componen participantes que según [71] pueden clasificarse en una de estas cinco categorías:

- 1.- Gestores superiores, que definen los aspectos de negocios que a menudo tienen una significativa influencia en el proyecto.
- 2.-Gestores (técnicos) del proyecto, que deben planificar, motivar, organizar y controlar a los profesionales que realizan el trabajo de software.
- 3.-Profesionales, que proporcionan las capacidades técnicas necesarias para la ingeniería de un producto o aplicación.
- 4.-Clientes, que especifican los requisitos para la ingeniería del software y otros elementos que tienen menor influencia en el resultado.
- 5.- Usuarios finales, que interaccionan con el software una vez que se ha entregado para la producción.

En la segunda P's, "producto", se deberían establecer estimaciones cuantitativas y un plan organizado, se debe establecer el ámbito del producto y delimitarlo debe ser unívoco y entendible a niveles de gestión y técnico. Los enunciados del ámbito del software deben estar delimitados. Es decir, los datos cuantitativos se establecen explícitamente; se anotan las limitaciones y se describen los factores de reducción de riesgos [71].

Tercera P's "proceso" en ella se proporciona la estructura desde la que se puede establecer un detallado plan para el desarrollo del software. Un pequeño número de

actividades estructurales se puede aplicar a todos los proyectos de software, sin tener en cuenta su tamaño o complejidad. Diferentes conjuntos de tareas permiten a las actividades estructurales adaptarse a las características del proyecto de Software y a los requisitos del equipo del proyecto [71].

En la cuarta P's, "proyecto", es donde surge el problema de seleccionar el modelo de proceso apropiado para la ingeniería del software que debe aplicar el equipo del proyecto [71].

El gestor del proyecto debe decidir qué modelo de proceso es el más adecuado para: 1) los clientes que han solicitado el producto y la gente que realizará el trabajo; 2) las características del producto en sí, y 3) el entorno del proyecto en el que trabaja el equipo de software. Cuando se selecciona un modelo de proceso, el equipo define entonces un plan de proyecto preliminar basado en un conjunto de actividades estructurales. Una vez establecido el plan preliminar, empieza la descomposición del proceso. Es decir, se debe crear un plan completo reflejando las tareas requeridas a las personas para cubrir las actividades estructurales [71].

Según [71], el personal se refiere a que debe organizarse en equipos eficaces, motivados para hacer un software de alta calidad y coordinados para alcanzar una comunicación efectiva. Los requisitos del producto deben comunicarse desde el cliente al desarrollador, dividirse (descomponerse) en las partes que lo constituyen y distribuirse para que trabaje el equipo de software. El proceso debe adaptarse al personal y al problema. Se selecciona una estructura común de proceso, se aplica un paradigma apropiado de ingeniería del software y se elige un conjunto de tareas para completar el trabajo. Finalmente, el proyecto debe organizarse de una manera que permita al equipo de software tener éxito.

Tabla 9.- Proceso de software (Ejemplos)

UN TRAJE	UNA GAMA DE AUTOMÓVILES	SOFTWARE
Personas: El sastre Producto: El traje Proyecto: La secuencia de acciones para hacer un traje concreto Proceso: Lo que aprende un sastre cuando aprende a hacer trajes	Personas: Empleados de la marca Producto: Los automóviles Proyecto: Desarrollo de un modelo nuevo Proceso: Las instrucciones de la empresa sobre cómo desarrollar un modelo nuevo	Personas: Nosotros Producto: La aplicación que utilizará Proyecto: IS1 Proceso: ???

Fuente: Elaboración propia [77]

5.3.3.1.- Modelo de Construcción de Prototipos

Descripción: Construcción de una implementación parcial que cubre los requisitos conocidos, para ir aprendiendo el resto y, paulatinamente, incorporarlos al sistema. Comienza con la recolección de requisitos luego aparece un «diseño rápido». El diseño rápido se centra en una representación de esos aspectos del software que serán visibles para el usuario/cliente, El diseño rápido lleva a la construcción de un prototipo. El prototipo lo evalúa el cliente/usuario y se utiliza para refinar los requisitos del software a desarrollar.

Ventajas

- Exige disponer de las herramientas adecuadas.
- No presenta calidad ni robustez.
- El cliente ve funcionando lo que para él es la primera versión del prototipo que ha sido construido con “plastilina y alambres”, y puede desilusionarse al decirle que el sistema aún no ha sido construido.
- El desarrollador puede caer en la tentación de ampliar el prototipo para construir el sistema final sin tener en cuenta los compromisos de calidad y de mantenimiento que tiene con el cliente.
- Puede ser demasiado lento, grande, torpe en su uso, o las tres a la vez.
- No hay otra alternativa que comenzar de nuevo.

Desventajas

- No modifica el flujo del ciclo de vida.
- Reduce el riesgo de construir productos que no satisfagan las necesidades de los usuarios.
- Reduce costos y aumenta la probabilidad de éxito [71].

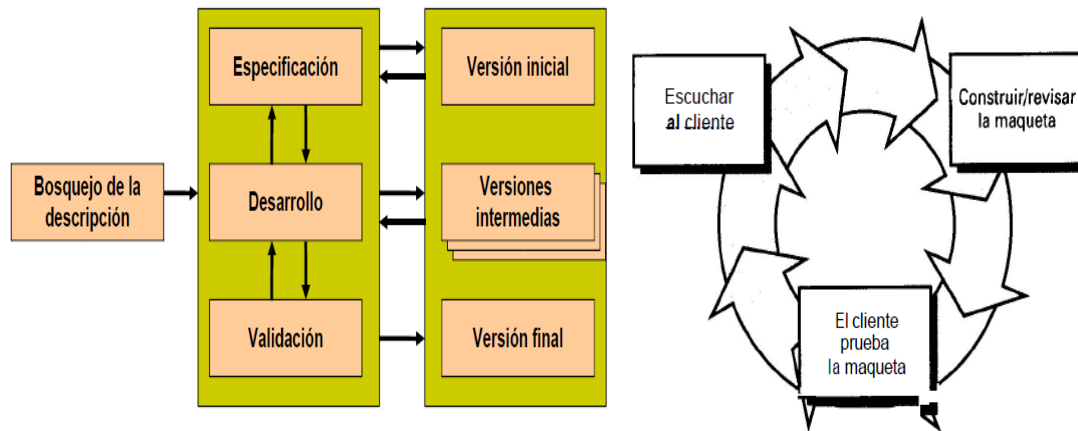


Figura 4 Modelo de construcción de prototipos Fuente [71]

5.3.3.2.- Modelos DRA

Descripción: El Desarrollo Rápido de Aplicaciones (DRA) es una adaptación a «alta velocidad» del modelo lineal secuencial en el que se logra el desarrollo rápido utilizando una construcción basada en componentes. Si se comprenden bien los requisitos y se limita el ámbito del proyecto, el proceso DRA permite al equipo de desarrollo crear un «sistema completamente funcional » Dentro de períodos cortos de tiempo.

Comprende las siguientes fases:

- 1.- Modelado de Gestión.
- 2.- Modelado de datos.
- 3.- Modelado del proceso.
- 4.- Generación de aplicaciones.
- 5.- Pruebas y entrega.

Ventajas

- Reduce tiempo de pruebas ya que enfatiza la reutilización, ya se han comprobado muchos de los componentes de los programas.
- Cada una de las funciones pueden ser afrontadas por un equipo DRA separado y ser integradas en un solo conjunto.

Desventajas

- Si no hay compromiso por ninguna de las partes constituyentes, los proyectos DRA fracasarán.
- Si un sistema no se puede modularizar adecuadamente, la construcción de los componentes necesarios para DRA será problemático.
- Si está en juego el alto rendimiento, y se va a conseguir el rendimiento convirtiendo interfaces en componentes de sistemas, el enfoque DRA puede que no funcione.
- No es adecuado cuando los riesgos técnicos son altos. Esto ocurre cuando una nueva aplicación hace uso de tecnologías nuevas, o cuando el software nuevo requiere un alto grado de interoperabilidad con programas de computadora ya existentes [54].

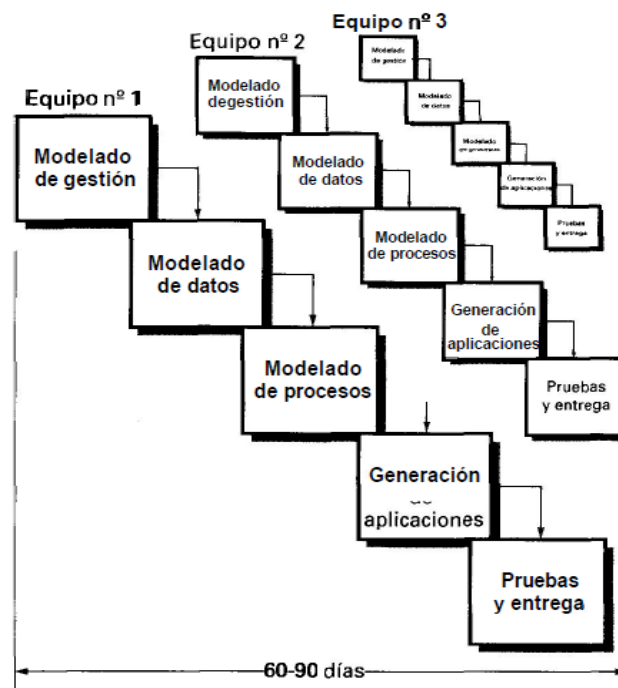


Figura 5 Modelo DRA Fuente [54]

5.3.3.3.- Modelo Incremental

Descripción: Combina elementos del modelo lineal secuencial (aplicados repetidamente) con la filosofía interactiva de construcción de prototipos.

Aplica secuencias lineales de forma escalonada mientras progresa el tiempo en el calendario. Cada secuencia lineal produce un «incremento» del software se centra en la entrega de un producto operacional con cada incremento.

Los primeros incrementos son versiones «incompletas» del producto final, pero proporcionan al usuario la funcionalidad que precisa y también una plataforma para la evaluación.

Ventajas:

- Se evitan proyectos largos y se entrega “Algo de valor” a los usuarios con cierta frecuencia.
- El usuario se involucra más.

Desventajas:

- Difícil de evaluar el coste total.
- Difícil de aplicar a sistemas transaccionales que tienden a ser integrados y a operar como un todo.
- Requiere gestores experimentados.
- Los errores en los requisitos se detectan tarde [53].

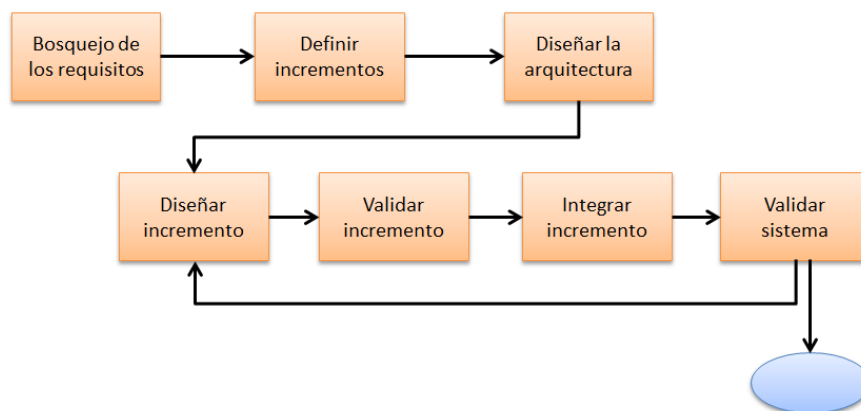


Figura 6 Modelo Incremental Fuente [53]

5.3.3.4.- Modelo Espiral

En la Ingeniería del Software, el paradigma del modelo en espiral propuesto por Boehm en 1988 se basa en la idea de desarrollo evolutivo. Es un enfoque recomendable para el desarrollo de software y de sistemas a gran escala. Este modelo tiene por finalidad mejorar u optimizar características tanto del ciclo de vida clásico, como de la creación de prototipos, añadiendo un nuevo elemento: el análisis de riesgo, otorgando al producto confiabilidad y utilidad [52] Figura 7.

El modelo espiral surge como un modelo no operativo de producción de software que tiende a poner énfasis allí donde los demás tienen sus debilidades, es decir, en el riesgo a asumir en cada etapa y el control del mismo [55].

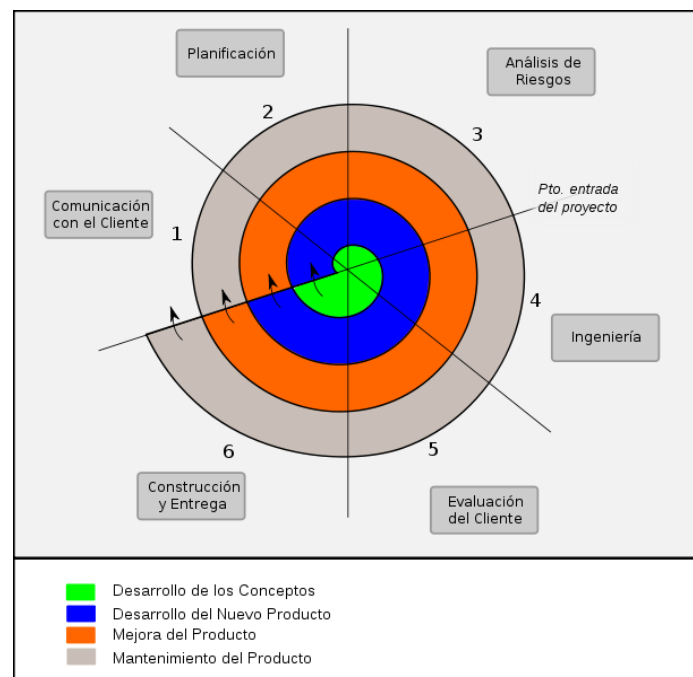


Figura 7 Modelo espiral Fuente [71]

El proceso del modelo espiral se compone de iteraciones (ciclo de prototipo iterativo). En cada ciclo, las nuevas expresiones o requerimientos obtenidos, transformando otras dadas, son examinadas a fin de determinar si representan progresos hacia el objetivo. La realimentación lograda en cada uno de estos ciclos

de desarrollo se plasma en el proceso dialéctico, donde los usuarios están en condiciones de definir e implementar más y mejores partes.

Paulatinamente el modelo se refina en cuanto a detalles e incorpora a través del riesgo cierta incertidumbre en los requisitos. En cada bucle alrededor del espiral, la culminación del análisis de riesgo resulta en una decisión de “seguir o no seguir” [52].

El modelo espiral basa sus características en sucesivas iteraciones hasta cumplir cierto hito o condiciones prefijadas para derivar a partir de allí en los modelos clásicos (cascada o prototipado) [55].

En cada vuelta o iteración hay que tener en cuenta

Los objetivos: Que necesidad debe cubrir el producto.

Alternativas: Las diferentes formas de conseguir los objetivos de forma exitosa, desde diferentes puntos de vista como pueden ser:

1. Características: experiencia del personal, requisitos a cumplir.
2. Formas de gestión del sistema.
3. Riesgo asumido con cada alternativa.

Desarrollar y Verificar: Programar y probar el software.

Si el resultado no es el adecuado o se necesita implementar mejoras o funcionalidades se comienza un nuevo ciclo de la espiral. La espiral tiene una forma de caracola y se dice que mantiene dos dimensiones, la radial y la angular:

1. Angular: Indica el avance del proyecto del software dentro de un ciclo.
2. Radial: Indica el aumento del coste del proyecto, ya que con cada nueva iteración se pasa más tiempo desarrollando.

Este sistema es muy utilizado en proyectos grandes y complejos como puede ser, por ejemplo, la creación de un Sistema Operativo. Al ser un modelo de Ciclo de Vida orientado a la gestión de riesgo se dice que uno de los aspectos fundamentales de su éxito radica en que el equipo que lo aplique tenga la necesaria experiencia y habilidad para detectar y catalogar correctamente los riesgos.

El modelo en espiral se divide conforme a [71] en un número de actividades de marco de trabajo, también llamadas regiones de tareas. Generalmente, existen entre tres y seis regiones de tareas.

- Comunicación con el cliente- las tareas requeridas para establecer comunicación entre el desarrollador y el cliente.
- Planificación- las tareas requeridas para definir recursos, el tiempo y otra información relacionadas con el proyecto.
- Análisis de riesgos- las tareas requeridas para evaluar riesgos técnicos y de gestión.
- Ingeniería- las tareas requeridas para construir una o más representaciones de la aplicación.
- Construcción y acción- las tareas requeridas para construir, probar, instalar y proporcionar soporte al usuario (por ejemplo: documentación y práctica).
- Evaluación del cliente- las tareas requeridas para obtener la reacción del cliente según la evaluación de las representaciones del software creadas durante la etapa de ingeniería e implementada durante la etapa de instalación.

Ventajas

- No requiere una definición completa de los requerimientos del software a desarrollar para comenzar su funcionalidad.
- En la terminación de un producto desde el final de la primera iteración es muy factible aprobar los requisitos.

- Sufrir retrasos corre un riesgo menor, porque se comprueban los conflictos presentados tempranamente y existe la forma de poder corregirlos a tiempo.

Desventajas

- Existe complicación cuando se evalúa los riesgos.
- Se requiere la participación continua por parte del cliente.
- Se pierde tiempo al volver producir inicialmente una especificación completa de los requerimientos cuando se modifica o mejora el software.

Cuando empieza este proceso evolutivo, el equipo de ingeniería del software gira alrededor de la espiral en la dirección de las agujas del reloj, comenzando por el centro. El primer circuito de la espiral puede producir el desarrollo de una especificación de productos; los pasos siguientes en la espiral se podrían utilizar para desarrollar un prototipo y progresivamente versiones más sofisticadas del software [71].

El modelo en espiral es un enfoque realista del desarrollo de sistemas y de software a gran escala. Como el software evoluciona, a medida que progresa el proceso el desarrollador y el cliente comprende y reaccionan mejor ante riesgos en cada uno de los niveles evolutivos.

El modelo en espiral utiliza la construcción de prototipos como mecanismo de reducción de riesgos, pero, lo que es más importante, permite a quien lo desarrolla aplicar el enfoque de construcción de prototipos en cualquier etapa de evolución del producto [55].

5.4.- Modelado (UML)

Después de haber reunido los requerimientos, [56] asume que se debe hacer un modelamiento del problema para obtener una visión más detallada del sistema y así poder resolver por medio del computador las diferentes situaciones que se presentan en el problema a tratar. En este proceso, es preferible trabajar con un

estándar de modelamiento, que sea compartido y comprendido por los analistas de diferentes sitios.

Para ello [58] señala dos filosofías de diseño las de caja blanca y caja negra. La caja negra se centra en la exploración de las distintas opciones del comportamiento externo a través de las entradas y salidas del programa y la de caja blanca se apoya en la exploración de los distintos elementos de la estructura interna. Los mejores resultados suelen aparecer combinando las ventajas de ambas aproximaciones.

Uno de los problemas de estas aproximaciones es la peculiaridad de los modelos empleados y que los analistas deben generar cuando realizan el trabajo de requerimientos de un sistema. Al tratarse de notaciones no convencionales que requieren un considerable esfuerzo, la aplicación de estas técnicas resulta poco viable en la mayoría de los entornos [58].

Frente a esta situación, [58] propone aproximarse a enfoques metodológicos con un mayor grado de implantación en los proyectos de desarrollo profesional y comercial. Así, la implantación de UML como notación de desarrollo orientado a objetos ha permitido la difusión e implantación real de técnicas de especificación de software.

[56] han definido un estándar de modelado denominado UML (Unified Modeling Language). UML está conformado por un conjunto de artefactos que permiten especificar los requisitos del software. La forma de representar a UML es conocida como metamodelo de UML. El metamodelo de UML sirve para que los ingenieros de Software puedan verificar la corrección de sus modelos, ya que tiene la estructura y las reglas que definen las interacciones entre los diferentes diagramas.

[56] señala que UML propone, entre otras cosas, un conjunto de diagramas que representan un software desde distintos puntos de vista. Los diagramas UML son independientes pero se conectan.

[63] menciona que el lenguaje UML tiene una notación gráfica muy expresiva que permite representar en mayor o menor medida todas las fases de un proyecto informático: desde el análisis con los casos de uso, el diseño con los diagramas de clases, objetos, entre otros, hasta la implementación y configuración con los diagramas de despliegue.

5.4.1.- Historia (UML)

El lenguaje UML comenzó a gestarse en octubre de 1994, cuando Rumbaugh se unió a la compañía Rational fundada por Booch. El objetivo de ambos era unificar dos métodos que habían desarrollado: el método Booch y el OMT (Object Modelling Tool). El primer borrador apareció en octubre de 1995. En esa misma época otro reconocido investigador, Jacobson, se unió a Rational y se incluyeron ideas suyas. Estas tres personas son conocidas como los “tres amigos”. Además, este lenguaje se abrió a la colaboración de otras empresas para que aportaran sus ideas. Todas estas colaboraciones condujeron a la definición de la primera versión de UML **[61]**.

Tal como indica su nombre, UML es un lenguaje de modelado. Un modelo es una simplificación de la realidad. El objetivo del modelado de un sistema es capturar las partes esenciales del sistema. Para facilitar este modelado, se realiza una abstracción y se plasma en una notación gráfica. Esto se conoce como modelado visual **[61]**.

5.4.2.- UML

[61] Establece que UML es un lenguaje que permite modelar, construir y documentar los elementos que forman un sistema software orientado a objetos.

Es un lenguaje de modelado que permite la representación conceptual y física de un sistema [62].

[61] Menciona los objetivos de UML como sigue:

Tabla 10 Objetivos de UML

Objetivo	Descripción
Visualizar	Permite expresar de una forma gráfica un sistema de forma que otro lo puede entender.
Especificar	Permite detallar cuáles son las características de un sistema antes de su construcción.
Construir	A partir de los modelos especificados se pueden construir los sistemas diseñados.
Documentar	Los propios elementos gráficos sirven como documentación del sistema desarrollado que pueden servir para su futura revisión.

Fuente. Elaboración propia, con datos de [61].

Aunque UML está pensado para modelar sistemas complejos con gran cantidad de software, el lenguaje es lo suficientemente expresivo como para modelar sistemas que no son informáticos, como flujos de trabajo (*workflow*) en una empresa, diseño de la estructura de una organización y por supuesto, en el diseño de hardware [63].

Según [61] un modelo UML está compuesto por tres clases de bloques de construcción:

- Elementos: Los elementos son abstracciones de cosas reales o ficticias (objetos, acciones, entre otros).
- Relaciones: relacionan los elementos entre sí.
- Diagramas: Son colecciones de elementos con sus relaciones.


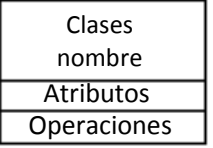
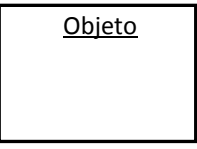
5.4.3.- Diagramas UML

Según [59] un diagrama es la representación gráfica de un conjunto de elementos con sus relaciones. En concreto, un diagrama ofrece una vista del sistema a

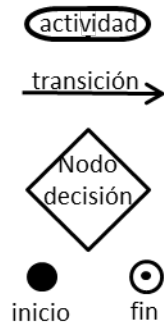
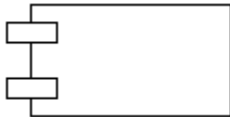
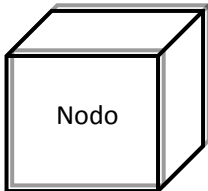
modelar. Para poder representar correctamente un sistema, UML ofrece una amplia variedad de diagramas para visualizar el sistema desde varias perspectivas. UML se puede usar para modelar distintos tipos de sistemas de: software, hardware, y organizaciones del mundo real.

UML prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan. Mientras que ha habido muchas notaciones y métodos usados para el diseño orientado a objetos, ahora los modeladores sólo tienen que aprender una única notación [59].

Tabla 11 Diagramas UML

Nombre del Diagrama	Descripción	Representación
Diagrama de casos de uso	Representa gráficamente cada interacción supuesta con el sistema a desarrollar, donde se representan los requisitos funcionales. Es decir, se está diciendo lo que tiene que hacer un sistema y cómo. Descripción de un conjunto de secuencias de acciones que ejecuta un sistema y que produce un resultado observable para un actor particular. Para modelar los procesos 'business'.	
Diagrama de clases	Muestra un conjunto de clases, interfaces y sus relaciones. Consiste de tres partes, divididas en compartimientos de nombres, atributos y operaciones. para modelar la estructura estática de las clases en el sistema.	
Diagrama de objetos	Los elementos objetos no tienen compartimientos. Los nombres de los objetos están subrayados y pueden mostrar el nombre del clasificador desde el cual el objeto se instancia (ubica). Notación similar a los diagramas de clases y se utilizan para ilustrar una instancia de una clase en un momento dado. Para modelar la estructura estática de los objetos en el sistema.	
Diagrama de	Muestra la interacción de un conjunto de	

secuencia	<p>objetos en una aplicación a través del tiempo y se modela para cada caso de uso. Definen acciones que se pueden realizar en la aplicación en cuestión. Existen dos tipos de mensajes: sincrónicos corresponden con llamadas a métodos del objeto que recibe el mensaje. Se representan con flechas con la cabeza llena y mensajes asíncrónicos terminan inmediatamente, y crean un nuevo hilo de ejecución dentro de la secuencia. Se representan con flechas con la cabeza abierta. También se representa con una flecha discontinua. para modelar el paso de mensajes entre objetos.</p>	
Diagrama de colaboración	<p>Modelan el comportamiento dinámico del sistema. Define una interacción y es una sociedad de roles y otros elementos que colaboran para proporcionar un comportamiento cooperativo mayor que la suma de los comportamientos de sus elementos. Se muestran como flechas etiquetadas unidas a los enlaces. Para modelar interacciones entre objetos.</p>	
Diagrama de estados	<p>Modelan el comportamiento dinámico del sistema, los estados se representan mediante óvalos. Las transiciones se representan mediante flechas con el nombre del evento respectivo. Se acostumbra poner un estado inicial (círculo negro). Es utilizado para identificar cada una de las rutas o caminos que puede tomar un flujo de información luego de ejecutarse cada proceso. Permite identificar bajo qué argumentos se ejecuta cada uno de los procesos y en qué momento podrían tener una variación. Para modelar el comportamiento de los objetos en el sistema.</p>	

Diagrama de actividades	<p>Modelan el comportamiento dinámico del sistema, representa el comportamiento interno de una operación o de un caso de uso, bajo la forma de un desarrollo por etapas, agrupadas secuencialmente. El propósito del diagrama de actividad es modelar el flujo de tareas y modelar las operaciones, para modelar el comportamiento de los Casos de Uso, objetos u operaciones.</p>	
Diagrama de componentes	<p>Enfocado a la implementación del sistema. Un componente es la parte física de un sistema, y se encuentra en la computadora, no en la mente del analista, por ejemplo una Tabla, un archivo de datos, biblioteca de vínculos dinámicos, documentos y cosas por el estilo. Se modela para que:</p> <ol style="list-style-type: none"> 1. Los clientes puedan ver la estructura del sistema finalizada. 2. Los desarrolladores cuenten con una estructura con la cual trabajar en adelante y 3. Quienes escriban las notas técnicas y la documentación puedan entender que escribirán para modelar componentes. 	
Diagrama de despliegue	<p>Enfocado a la implementación del sistema. Muestran las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación. Un nodo es un recurso de ejecución tal como un computador, un dispositivo o memoria. Instancias o tipos de nodos que se representa como un cubo 3D. Para modelar la distribución del sistema.</p>	

Fuente Elaboración propia, con datos de [61] y [62].

5.4.4.- Proceso Unificado de Desarrollo

Aunque UML es bastante independiente del proceso de desarrollo que se siga, los mismos creadores de UML han propuesto su propia metodología de desarrollo, denominada el Proceso Unificado de Desarrollo [61].

El Proceso Unificado está basado en componentes, lo cual quiere decir que el sistema software en construcción está formado por componentes software interconectados a través de interfaces bien definidos. Además, el Proceso Unificado utiliza UML para expresar gráficamente todos los esquemas de un sistema software [61].

Los aspectos que definen este Proceso Unificado son tres: iterativo e incremental, dirigido por casos de uso y centrado en la arquitectura [61].

- Dirigido por casos de uso. Basándose en los casos de uso, los desarrolladores crean una serie de modelos de diseño e implementación que los llevan a cabo. Además, estos modelos se validan para que sean conformes a los casos de uso. Finalmente, los casos de uso también sirven para realizar las pruebas sobre los componentes desarrollados.
- Centrado en la arquitectura. En la arquitectura de la construcción, antes de construir un edificio éste se contempla desde varios puntos de vista: estructura, conducciones eléctricas, fontanería, entre otros. Cada uno de estos aspectos está representado por un gráfico con su notación correspondiente. Siguiendo este ejemplo, el concepto de arquitectura software incluye los aspectos estáticos y dinámicos más significativos del sistema.
- Iterativo e incremental. Todo sistema informático complejo supone un gran esfuerzo que puede durar desde varios meses hasta años. Por lo tanto, lo más práctico es dividir un proyecto en varias fases. Actualmente se suele hablar de ciclos de vida en los que se realizan varios recorridos por todas las fases. Cada recorrido por las fases se denomina iteración en el proyecto en la que se

realizan varios tipos de trabajo (denominados flujos). Además, cada iteración parte de la anterior incrementado o revisando la funcionalidad implementada. Se suele denominar proceso.

Como ya se observó UML define varios tipos de diagramas que se utilizan para describir diferentes aspectos o vistas de un sistema. En particular, los diagramas de Casos de Uso que se han adoptados casi universalmente para capturar los requerimientos de los sistemas de software; sin embargo, los Casos de Uso son más que una herramienta de especificación ya que tienen una gran influencia sobre todas las fases del proceso de desarrollo tales como el diseño, la implementación y las pruebas del sistema [60].

Los diagramas más interesantes (y los más usados) son los de casos de uso, clases y secuencia, [59].

5.4.5.- Casos de uso

Propuestos por Jacobson los Casos de Uso son una parte esencial de la notación UML para la especificación del comportamiento externo de un sistema. Un Caso de Uso se define como “unidad coherente de funcionalidad manifestada por secuencias de mensajes intercambiados por el sistema y uno o más actores externos junto a las acciones desarrolladas por el sistema”. Supone una buena expresión de la interacción de actores con el sistema para obtener un resultado observable y valioso para los mismos [58].

Un diagrama de Casos de Uso muestra la funcionalidad que ofrece el sistema futuro desde la perspectiva de los usuarios externos al mismo [56].

A partir de la publicación de Jacobson, gran parte de los más reconocidos especialistas en métodos Orientados a Objetos coincidieron en considerar a los Casos de Uso como una excelente forma de especificar el comportamiento externo de un sistema. De esta forma, la notación de los Casos de Uso fue incorporada al

lenguaje estándar de modelado UML y avalado por las principales empresas que desarrollan software en el mundo [95].

El modelo de Casos de Uso es un diagrama que describe gráficamente quien va a usar el sistema y de qué manera el usuario espera interactuar con él. La modelación de los Casos de Uso identifica y describen las funciones del sistema [72].

[95] menciona un Caso de Uso es una secuencia de interacciones entre un sistema y alguien o algo que usa alguno de sus servicios.

Según [96] los elementos de un modelo de Casos de Uso son:

- ❖ Casos de Uso
- ❖ Actores
- ❖ Relaciones

Los Casos de Uso se representan gráficamente con una elipse horizontal con el nombre del caso que aparece encima, debajo o dentro de la elipse. Un Caso de Uso representa un objetivo individual del sistema y describe una secuencia de actividades y de interacciones del usuario para tratar de alcanzar el objetivo [72].

Los Casos de Uso se emplean para capturar la esencia de los problemas de negocio y para modelar la funcionalidad del sistema. Adicionalmente son un punto inicial para la identificación de las entidades de datos o de los objetos del sistema. Y como los Casos de Uso contienen una enorme cantidad de detalle de la funcionalidad del sistema, estos serán un recurso constante para validar al sistema [72].

Es importante tener clara la diferencia entre usuario y actor. Un “Actor” es una clase de rol, mientras que un usuario es una persona que, cuando usa el sistema, asume un rol. De esta forma, un usuario puede acceder al sistema como distintos actores.

La forma más simple de entender esto es pensar en perfiles de usuario de un sistema operativo. Una misma persona puede acceder al sistema con distintos perfiles, que le permiten hacer cosas distintas. Los perfiles son en este caso equivalentes a los actores [95].

Por ende los Casos de Uso se inician o son generados por los usuarios externos llamados actores. Según [72] “Actor” es cualquier cosa que necesita interactuar con el sistema para intercambiar información. Y [97] describe “Actor” como roles que juegan los usuarios con respecto al sistema.

Utilizaremos el ejemplo de un empleado que ingresa a trabajar a una empresa. El “Actor” sería el empleado y el Caso de Uso sería ingreso a la empresa. Un “Actor” representa un papel desempeñado por un usuario que interactúa con el sistema y no significa que retrate a una persona o un puesto de trabajo. De hecho un “Actor” no tiene que ser humano. Puede ser una organización, otro sistema de información, un dispositivo externo, entre otros [72].

Un “Actor” se representa gráficamente como una figura de línea rotulada con el nombre del papel que juega el actor, se representan con dibujos simplificados de personas, llamados en inglés “stick man” (hombres de palo) [95].

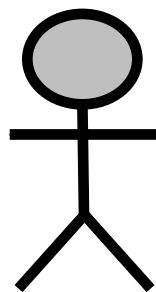


Figura 8 Símbolo de un actor. Fuente: [72]

Es importante observar que hay tipos de “Actores”:

Tabla 10.- Tipos de “Actores”

Tipo de Actor	Descripción
Actor primario de negocios	El interesado que se beneficia de la ejecución de un caso de uso al recibir algo de valor medible u observable. Por ejemplo un empleado que recibe un cheque como pago del sistema de nómina cada quincena, el empleado no inicia el evento pero es el receptor primario del algo de valor.
Actor primario del sistema	El involucrado que tiene una interfaz directo con el sistema para iniciar u ocasionar el evento de negocios o del sistema. Estos actores facilitan el evento a través del uso directo del sistema para beneficio del actor primario de negocios. Por ejemplo un analista de recursos humanos que captura los datos del empleado para que este ingrese a la empresa.
Actor externo servidor	El involucrado que responde a una solo solicitud desde el caso de uso. Por ejemplo un buró de crédito que autoriza el pago mediante tarjeta de crédito.
Actor externo receptor	El involucrado que no es el actor primario pero que recibe algo de valor medible u observable proveniente del caso de uso. Por ejemplo un equipo de cómputo que recibe una orden de mantenimiento después de que un empleado ha colocado una solicitud.

Fuente: Elaboración propia, con información de [72].

Relaciones. Una relación se ilustra como una línea entre dos símbolos en el diagrama de Casos de Uso. El significado de las relaciones puede diferir dependiendo de cómo se dibujen las líneas y que tipos de símbolos conectan. Ahora definiremos los tipos de relaciones que se encuentran en un diagrama de Casos de Uso [72].

Según [96] entre Casos de Uso pueden darse relaciones como:

- Asociación
- Extensión
- Usos o inclusión
- Dependencia
- Herencia

Asociación. Es la relación entre un “Actor” y un Caso de Uso en la que interactúan entre sí. Como se indica en la Figura 9. Una asociación se modela como una línea continua que conecta al “Actor” y al Caso de Uso. Una asociación que conecta una cabeza de flecha en el extremo que toca al Caso de Uso *1* indica que el caso fue iniciado por el “Actor” en el otro extremo de la línea. Las asociaciones sin cabeza de flecha *2* indican una iteración entre el Caso de Uso y un “Actor” externo servidor o receptor. Sin un “Actor” se asocia con un Caso de Uso, decimos que el “Actor” se comunica con el caso. Las asociaciones pueden ser bidireccionales o unidireccionales.

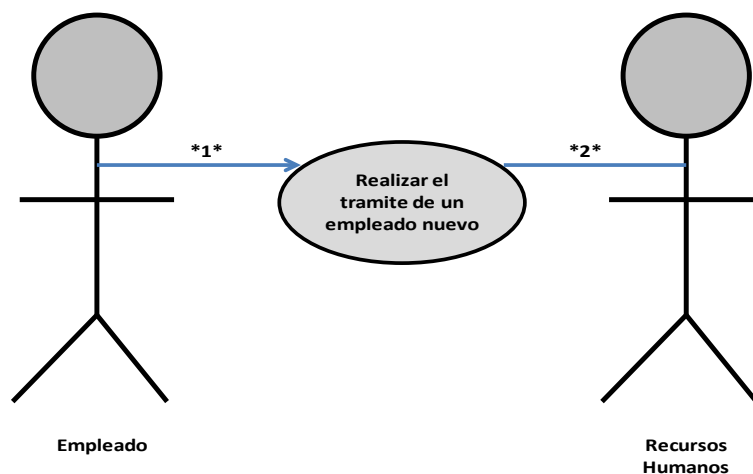


Figura 9 Asociación Casos de Uso Fuente [72]

Extensión. Un Caso de Uso puede contener una funcionalidad compleja que consiste de varios pasos que hacen difícil entender a la lógica del caso. Con objeto de simplificar el Caso de Uso y hacer que se entienda más fácilmente, se extraen los pasos más complejos para formar un propio caso. El caso resultante se llama un Caso de Uso de extensión ya que extiende la funcionalidad del Caso de Uso original. La relación entre el Caso de Uso de extensión y el que se está extendiendo se llama una relación de extensión. Un Caso de Uso puede tener muchas relaciones de extensión, pero un Caso de Uso de extensión puede ser invocado solamente por el caso que se esté extendiendo. Como se muestra en la Figura 10,

la relación de extensión se representa como una línea con cabeza de flecha que comienza en el Caso de Uso de extensión y que apunta al Caso de Uso que se está extendiendo. Cada línea de relación de extensión se rotula como <<extensión>>. Generalmente los Casos de Uso de extensión no se identifican en la fase de requerimientos, sino en la de análisis [72].

La relación *extend* establece que un Caso de Uso puede ser extendido con algún comportamiento adicional definido en otro Caso de Uso. La relación contiene una condición y referencia una secuencia de puntos de extensión en el Caso de Uso base. Una vez que una instancia del Caso de Uso ejecuta un comportamiento referenciado por el primer punto de extensión de la relación, la condición es evaluada. Si se cumple, la secuencia de la instancia se extiende para incluir la secuencia del Caso de Uso extensión [60].

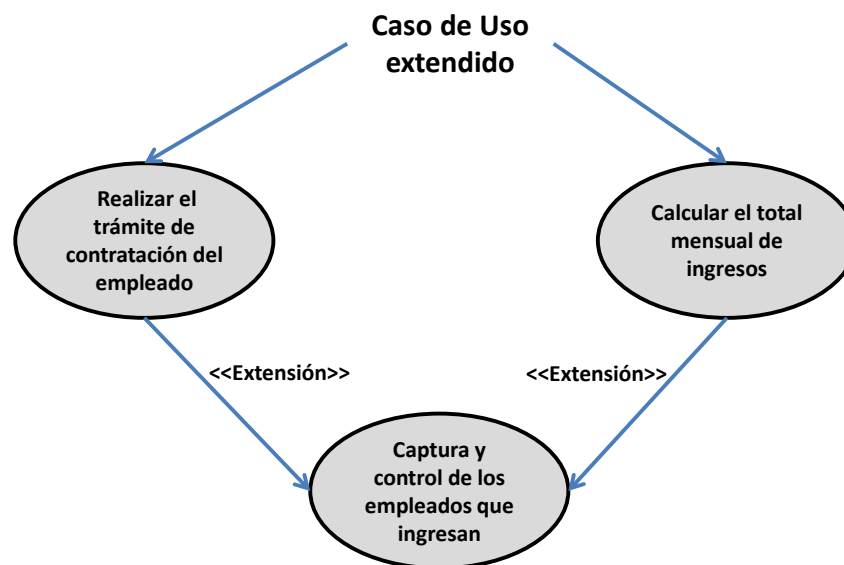


Figura 10 Extensión Casos de Uso. Fuente. Elaboración propia con datos de [72]

Usos o inclusión. Muy comúnmente, se pueden descubrir dos o más Casos de Uso que ejecuten pasos de funcionalidad idéntica. Lo mejor es extraer estos comunes para formar un Caso de Uso separado que sea propio llamado un Caso de Uso resumen. Un Caso de Uso resumen representa una forma de “reusó” y es una

herramienta excelente para reducir la redundancia entre los Casos de Uso. Un Caso de Uso resumen está disponible como referencia (o uso) para cualquier otro Caso de Uso que requiera su funcionalidad. La relación entre el Caso de Uso resumen y el Caso de Uso que lo usa se llama una relación de uso. La relación de uso que se presenta en la Figura 11 se ilustra con una línea con cabeza de flecha que comienza en el Caso de Uso oficial y que apunta al Caso de Uso que esté usando. Cada línea de relación de uso se rotula con `<<uso>>` [72].

Una relación incluye entre dos Casos de Uso indica que el comportamiento definido en el Caso de Uso a adicionar, es incluido en un lugar dentro de la secuencia del comportamiento realizado por una instancia del Caso de Uso base. Cuando una instancia del Caso de Uso «llega al lugar» donde el comportamiento de otro Caso de Uso debe ser incluido, ejecuta todo el comportamiento descrito por el Caso de Uso incluido y luego continúa de acuerdo a su Caso de Uso original. El Caso de Uso incluido no depende del Caso de Uso base. En este sentido, el Caso de Uso incluido representa comportamiento encapsulado que puede ser reusado en varios Casos de Uso [60].

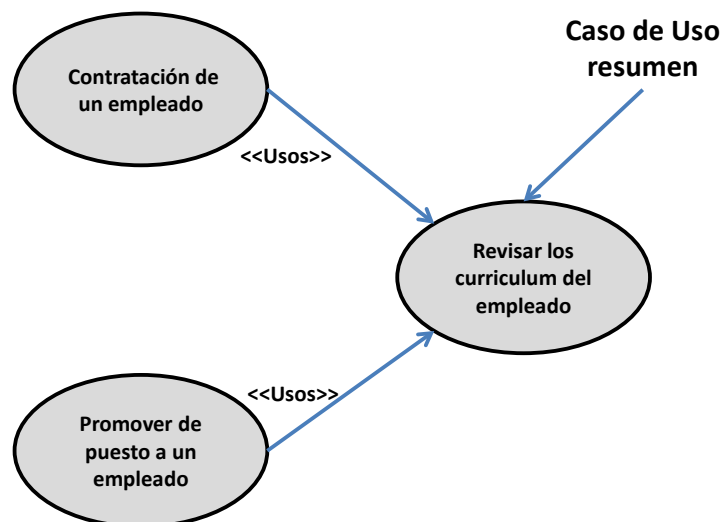


Figura 11 Usos Casos de Uso. Fuente. Elaboración propia con información de [72]

Dependencia. Es de mucha ayuda saber cuáles Casos de Uso tienen una dependencia sobre otros Casos de Uso con objeto de determinar la secuencia en que es necesario desarrollar los Casos de Uso. Por ejemplo, cuando un empleado no puede subir de puesto hasta que haya ocurrido el Caso de Uso de tramitar la promoción. Un diagrama de Casos de Uso que modele las dependencias de Caso de Uso del sistema mediante el uso de la relación de dependencia proporciona un modelo que es una herramienta excelente para propósitos de planeación y de programación. La relación de dependencia tal como se presenta en la Figura 12 se ilustra con una línea con cabeza de flecha que comienza en un Caso de Uso y que apunta al caso del cual depende. La línea de relación de dependencia se rotula <<depende de>> [72].

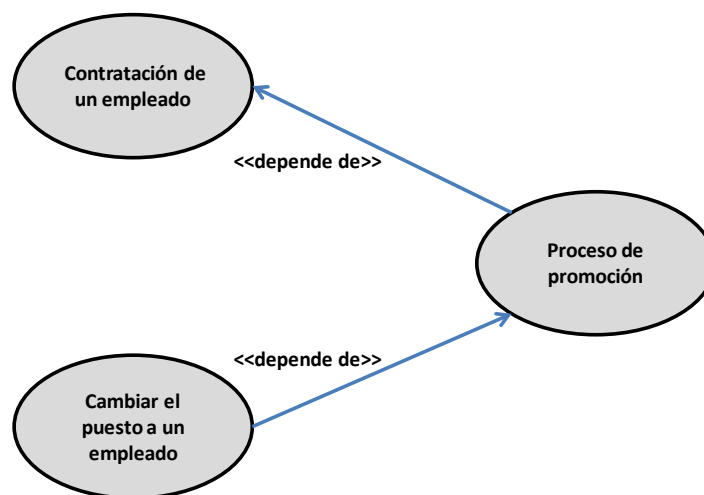


Figura 12 Dependencia Casos de Uso. Fuente. Elaboración propia con datos de [72]

Herencia. Cuando dos o más actores pueden iniciar el mismo Caso de Uso lo mejor es extrapolar este comportamiento común y asignarlo a un nuevo “Actor” resumen con objeto de reducir la comunicación redundante con el sistema [72].

Una relación de generalización entre Casos de Uso implica que el Caso de Uso hijo hereda todos los atributos, secuencias de comportamiento, puntos de extensión y relaciones definidos en el Caso de Uso padre. El Caso de Uso hijo puede definir

nuevas operaciones, como también redefinir o enriquecer con nuevas secuencias de acciones operaciones ya existentes en el Caso de Uso padre. Para distinguir si la especialización está redefiniendo una operación del padre o agregándole secuencias de acciones, sugerimos la inclusión de un estereotipo (elemento de UML) <<redefine>> para el primer caso o <<enrichment>> para el segundo, en la operación en cuestión [60].

Los diagramas de actividades (DA) son parte de los diagramas de comportamiento UML, que describen la funcionalidad del software en un nivel alto de abstracción.

El Caso de Uso como elemento principal de los ejes del trazado se relaciona con predecesores y sucesores por medio de las relaciones de trazado <<refine>> y <<realice>>, lo que permite obtener transitividad, entre otros elementos, de ejes, predecesores y sucesores [57].

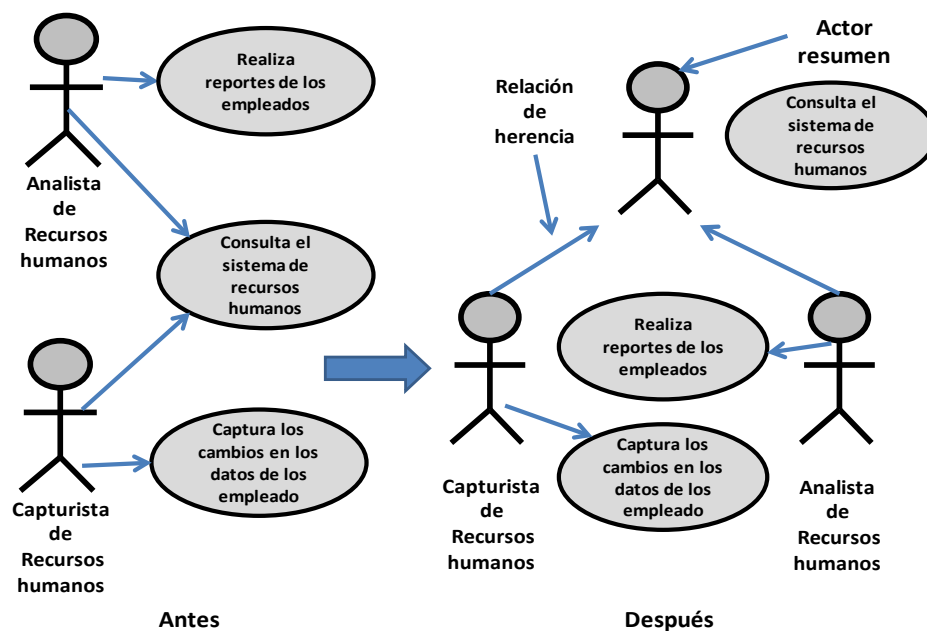


Figura 13 Herencia Casos de Uso. Fuente. Elaboración propia con datos de [72]

Los Casos de Uso, son útiles en discusiones con usuarios finales para asegurar que hay concordancia con los requerimientos realizados sobre el sistema, sobre sus

limitaciones, entre otros. Más precisamente, un Caso de Uso especifica un conjunto de secuencias completas de acciones que el sistema puede realizar. Cada secuencia es iniciada por un usuario del sistema. Esto incluye la interacción entre el sistema y su entorno como también la respuesta del sistema a estas interacciones [60].

5.5.- Base de datos

Una *base de datos* es un conjunto de datos almacenados en memoria externa que están organizados mediante una estructura de datos. Cada base de datos ha sido diseñada para satisfacer los requisitos de información de una empresa u otro tipo de organización [45], Una base bien diseñada facilita la administración de datos y se convierte en un valioso generador de información, de lo contrario tiende a generar errores que probablemente conduzcan a malas decisiones; pero con el tiempo pueden ser autocorregible [44].

Una base de datos se puede percibir como un gran almacén de datos que se define y se crea una sola vez, y que se utiliza al mismo tiempo por distintos usuarios [45].

Según [46] las bases de datos son una herramienta de vital importancia para el desarrollo de la actividad profesional, considera como base de datos cualquier recopilación organizada de información sobre la que haya habido análisis documental y que disponga de un sistema de búsqueda específico.

Las bases de datos son componentes esenciales de muchos sistemas de información. El papel de una base de datos es proporcionar una memoria de largo plazo contiene entidades y relaciones. Las bases de datos no son los únicos componentes de los sistemas de información; también contienen personas, procedimientos, datos de entrada, datos de salida, software y hardware. Por lo tanto, el desarrollo de un sistema de información involucra más que el simple desarrollo de una base de datos [8].

Tabla 11.- Tipos de bases de datos

Estructura jerárquica	<p>Fue la primera en utilizarse, a finales de los años sesentas del siglo XX, y se enfocaba sobre todo en el establecimiento de jerarquías o niveles entre los distintos campos de la base de datos, basándose en el criterio de que los campos de mayor jerarquía son los más genéricos. De ahí que su estructura sea arborescente y que para acceder a un campo que se encuentre en cualquier nivel, es necesario localizarlo partiendo del nivel superior; es decir, se maneja una organización de Padre-Hijo.</p> <p>Ventaja: Es su facilidad y simplicidad. Desventaja: Lentitud.</p>
Estructura de red	<p>Es una estructura donde existe más de una conexión o enlace entre los campos de la base de datos, de tal manera que se pueda acceder a los datos desde diferentes localidades, gracias a la interconexión de los datos entre los distintos niveles, sin la necesidad de iniciar con los campos de mayor jerarquía.</p> <p>Este modelo nace para resolver la lentitud de la organización jerárquica, permitiendo múltiples relaciones Padre-Hijo.</p> <p>Ventajas: Su rapidez. Desventaja: Es la rigidez ya que una vez creada la estructura es imposible su modificación.</p>
Estructura relacional	<p>Se caracteriza porque su estructura no se basa en campos, sino en las tablas en su conjunto; es decir, la información se organiza por categoría de datos, en las cuales hay campos en común entre ellas y forman lo que se conoce como relaciones.</p>
Estructura orientada a objetos	<p>Estas bases agrupan los elementos de datos y sus características, ya que define sus atributos y procedimientos asociados en elementos completos llamados objetos en donde un objeto puede ser cualquier cosa, un color, una casa, un producto u otros, y se diseñan con el objetivo de trabajar en conjunción con lenguajes de programación orientados a objetos.</p>

Fuente: Elaboración propia con información de [1].

Tabla 12.- Definiciones y conceptos de bases de datos

Datos	Conjunto de caracteres con algún significado, pueden ser numéricos, alfabéticos, o alfanuméricos, cualquier tipo de carácter sin ser procesado, también lo podemos definir como una unidad mínima de información.
Banco de datos	Colección de datos interrelacionados almacenados y procesables de forma independiente de los archivos o programas estructurados que los utilizan. Sistemas creados para almacenar un conjunto de datos del mismo tipo, de modo que se puedan actualizar y utilizar en cualquier momento y ocupen el menor espacio posible.
Información	Es el resultado obtenido de la transformación de los datos; conjunto ordenado de datos procesados, los cuales son manejados con un fin determinado o para satisfacer un requerimiento específico, según la necesidad del usuario. Para que un conjunto de datos pueda ser procesado eficientemente y pueda dar lugar a la información, primero se debe guardar lógicamente en archivos.
Campo	Un campo es una pieza única de información, también podemos decir que es la unidad más pequeña a la cual uno puede referirse en un programa. Desde el punto de vista del programador representa una característica de un individuo u objeto.
Atributos	Son las características básicas que identifican a la entidad (campo de datos).
Entidad	Una entidad es una persona, un lugar, una cosa, un evento o un concepto acerca del cual se desea registrar información. Ejemplos de entidades son clientes, proveedores, inventarios, empleados, otro.
Registro	Un registro es un sistema completo de campos, es decir es una colección de datos relacionados, referentes a una entidad. En otras palabras el renglón completo de una tabla o tupla.
Archivo	Un archivo es una colección de registros almacenados siguiendo una estructura homogénea. Por ejemplo, una guía de teléfono es análoga a un archivo. Contiene una lista de registros, cada uno de los cuales consiste en tres campos: nombre, dirección, y número de teléfono.
Archivos de datos	Un conjunto de registros forman un archivo de datos.

Fuente: Elaboración propia con información de [44].

5.5.1.- Ventajas de las bases de datos

En la actualidad una base de datos se ha convertido en un recurso indispensable para tener un control y organización de grandes cantidades de información.

Ventajas del enfoque de bases de datos [43]:

- Disminuye la redundancia.
- Evita la inconsistencia.
- Conserva la integridad de los datos.
- Permite compartir datos en diferentes aplicaciones.
- Facilita el desarrollo de aplicaciones.
- Uniformiza los controles de seguridad, privacidad e integridad.
- Proporciona independencia entre los datos y los programas.
- Reducción en el mantenimiento a programas.
- La información se encuentra centralizada.
- Usando una BD se pueden aplicar restricciones en el acceso a los datos por motivos de seguridad.
- El acceso puede ser en línea o batch.
- Reduce el tiempo de desarrollo y mantenimiento de las aplicaciones.
- Favorece el cumplimiento de normas para representación de los datos.
- Otras ventajas: Más eficiente gestión de almacenamiento.

5.5.2- Modelo relacional

Evolución del modelo relacional

Tabla 13.- Evolución del modelo relacional

AÑO	DESCRIPCIÓN
1970	Codd publica las bases del modelo relacional
1971-72	Primeros desarrollos teóricos
1973-78	Primeros prototipos
1978	Aparece el lenguaje QBE
1979	Aparece Oracle

1980	Aparece Ingres
1981	Aparece SQL
1982	Aparece DB2
1986	ANSI normaliza el SQL (SQL/ANSI)
1987	SQL de ISO
1990	Versión dos del modelo relacional (RM/V2)
1992	SQL 92
1998	SQL 3

Fuentes: Elaboración propia con información de [47].

El modelo relacional representa la base de datos como una colección de relaciones [41].

Edgar Frank Codd definió las bases del modelo relacional a finales de los 60. Trabajaba para IBM empresa que tardó un poco en implementar sus bases. Pocos años después el modelo se empezó a implementar cada vez más, hasta ser el modelo de bases de datos más popular [47].

En las bases de Codd se definían los objetivos de este modelo [47]:

- Independencia física. La forma de almacenar los datos, no debe influir en su manipulación lógica.
- Independencia lógica. Las aplicaciones que utilizan la base de datos no deben ser modificadas por que se modifiquen elementos de la base de datos.
- Flexibilidad. La base de datos ofrece fácilmente distintas vistas en función de los usuarios y aplicaciones.
- Uniformidad. Las estructuras lógicas siempre tienen una única forma conceptual (las tablas).
- Sencillez.

El modelo relacional utiliza tablas que proporcionan esta interfaz, donde el modelo de datos se especifica en una serie de tablas o relaciones [42].

Una base de datos relacionales consiste en un conjunto de tablas, a cada una de las cuales se le asigna un nombre exclusivo [48].

5.5.3.- Tablas

Las bases de datos relacionales se basan en el uso de tablas (también se las llama relaciones). Las tablas se representan gráficamente como una estructura rectangular formada por filas y columnas. Cada columna almacena información sobre una propiedad determinada de la tabla, se le llama atributo, nombre, apellidos, edad, entre otros.

Cada fila posee una ocurrencia o ejemplar de la instancia o relación representada por la tabla, a las filas se las llaman tuplas [47].

- Atributo: Cada columna de la tabla.
- Tupla: Cada fila de la tabla (cada ejemplar que la tabla representa).
- Registro: Es cada una de las filas de la tabla, y está formado por el dato de cada uno de los campos almacenados en una misma operación.
- Dominio: Conjunto válido de valores representables por un atributo.

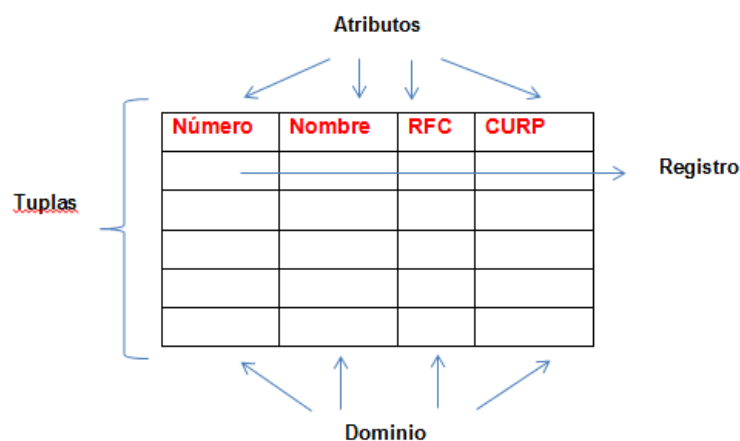


Figura 14 Partes de una tabla Fuente [47].

De acuerdo con [47] se tienen los siguientes tipos de tablas:

- a) Persistentes. Sólo pueden ser borradas por los usuarios:
- Base. Independientes, se crean indicando su estructura y sus ejemplares.
 - Vistas. Son tablas que sólo almacenan una definición de consulta, resultado de la cual se produce una tabla cuyos datos proceden de las bases o de otras vistas e instantáneas. Si los datos de las tablas base cambian, los de la vista que utiliza esos datos también cambia.
 - Instantáneas. Son vistas (creadas de la misma forma) que sí que almacenan los datos que muestra, además de la consulta que dio lugar a esa vista. Sólo modifican su resultado (actualizan los datos) siendo refrescadas por el sistema cada cierto tiempo.
- b) Temporales. Son tablas que se eliminan automáticamente por el sistema. Pueden ser de cualquiera de los tipos anterior,

[43] denomina llave (o clave) al atributo que permite significar de manera única a una entidad. Una llave es en otras palabras, el campo a partir del cual se pueden inferir otros campos de una tabla, por lo que, cada tupla debe estar asociada con una llave que permita su identificación.

Tipos de llaves

De acuerdo con **[43]** los tipos de llaves son:

- Llave primaria: Esta llave debe ser un atributo único e irrepetible de tal modo que se pueda hacer referencia a una tabla por medio de esta.

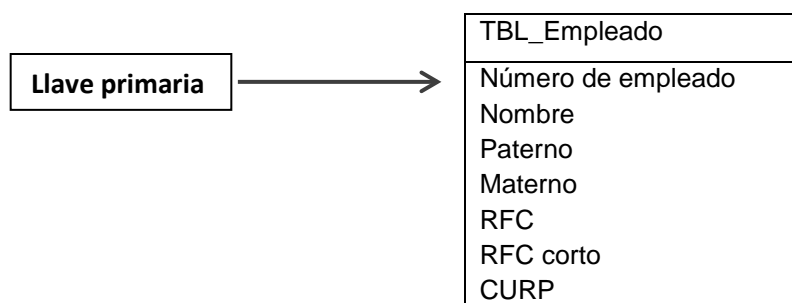


Figura 15 Ejemplo de Llave primaria Fuente elaboración propia con información de [43]

- Llave foránea: es una columna o conjunto de columnas que se refieren a una clave primaria de la misma tabla a otra. Se crean estas claves para reforzar las reglas de diseño de la base de datos relacionales. Una tabla puede tener más de una llave primaria.

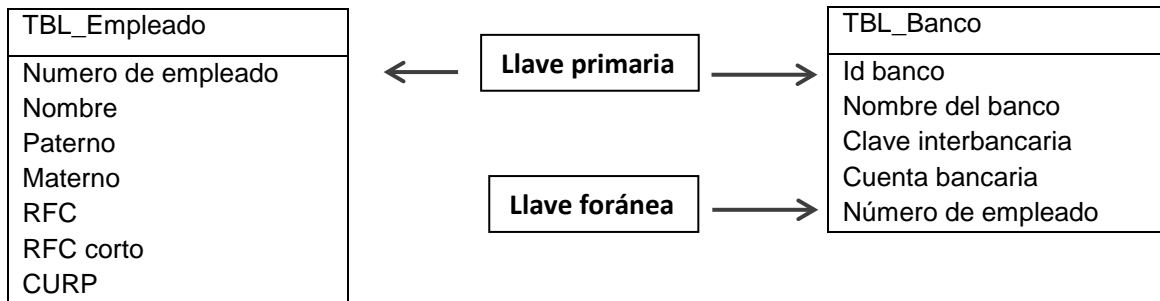


Figura 16 Ejemplo de Llave Foránea Fuente elaboración propia con información de [43]

5.5.4.- Esquema entidad relación

En principio las entidades fuertes del modelo Entidad Relación son transformados al modelo relacional siguiendo estas instrucciones:

- Entidades. Las entidades pasan a ser tablas
- Atributos. Los atributos pasan a ser columnas.
- Identificadores principales. Pasan a ser claves primarias
- Identificadores candidatos. Pasan a ser claves candidatas.

5.5.5.- Relaciones

[40] menciona que el propósito fundamental de las relaciones es evitar tener datos dobles o duplicados, es decir, tener redundancia en la información. La redundancia es el almacenamiento de los mismos datos más de una vez en diferentes lugares lo que provoca problemas como:

- Incremento de tiempo
- Incremento de trabajo

- Desperdicio de datos
- Inconsistencia de datos

Tipos de relaciones entre tablas

Tabla 14.- Tipos de relaciones entre tablas

Relaciones “Uno a varias”	Tipo más frecuente de relación, en esta relación un registro de la tabla A puede tener más de un registro coincidente en la tabla B, pero un registro de la tabla B tiene como máximo un registro coincidente en la tabla A.
Relaciones “Varios a varios”	Un registro de la tabla A puede un registro coincidente en la tabla B y viceversa, para detectar esta relación obsérvela en dos sentidos, por ejemplo: examine la relación pedidos y productos. Un pedido puede incluir más de un producto y un producto puede aparecer en varios pedidos, para resolver este problema la regla a seguir es crear una tercera tabla que rompa esta relación que incluiría la clave principal de cada una de las dos tablas anteriores como clave principal en esta tabla que garantiza un valor único para cada registro.
Relación “Uno a uno”	Un registro de la tabla A no puede tener más de un registro coincidente en la tabla B y viceversa. Este tipo de relación es poco frecuente y ocurre al aplicar técnicas de desnormalización, cuando una tabla es muy grande y se divide en dos, o cuando la información de algunos campos tiene un tiempo de vida útil limitado o cuando esa misma información solo la utilizan un grupo limitado de registros.

Fuente: Elaboración propia con información de [12]

Problemas del esquema relacional

Una vez obtenido el esquema relacional resultantes del modelo entidad relación que representaba la base de datos, normalmente tendremos una buena base de datos. Pero otras veces, debido a fallos en el diseño o a problemas indetectables en esta fase del diseño, tendremos un esquema que puede producir una base de datos que incorpore estos problemas [47]:

- Redundancia. Se llama así a los datos que se repiten continua e innecesariamente por las tablas de las bases de datos.

- Ambigüedades. Datos que no clarifican suficientemente el registro al que representan.
- Pérdida de restricciones de integridad.
- Anomalías en operaciones de modificación de datos. El hecho de que al insertar un solo elemento haya que repetir tuplas en una tabla para variar unos pocos datos. O que eliminar un elemento suponga eliminar varias tuplas.

El principio fundamental reside en que las tablas deben referirse a objetos o situaciones muy concretas. Lo que ocurre es que conceptualmente es difícil obtener ese problema.

La solución suele ser dividir la tabla con problemas en otras tablas más adecuadas.

5.5.6.- Normalización

Es el proceso mediante el cual se transforman datos complejos a un conjunto de estructuras de datos más pequeñas. También se puede entender la normalización como una serie de reglas que sirven para ayudar a los diseñadores de bases de datos a desarrollar un esquema que minimice los problemas de lógica.

Una ventaja de la normalización de base de datos es el consumo de espacio. Una base de datos normalizada ocupa menos espacio en disco que una no normalizada hay menos repetición de datos **[70]**.

Las bases de datos relacionales se normalizan para **[43]**:

- Evitar la redundancia de los datos.
- Evitar problemas de actualización de los datos en las tablas.
- Proteger la integridad de los datos.

Ventajas de la normalización son las siguiente **[43]**:

- Evita anomalías en inserciones, modificaciones y borrados.

- Mejora de la independencia de datos.
- No establece restricciones artificiales en la estructura de los datos.

Existen básicamente tres niveles de normalización:

- Primera forma normal (1FN)
- Segunda forma normal (2FN)
- Tercera forma normal (3FN)

Primera forma normal (1FN)

Establece que las columnas repetidas deben eliminarse y colocarse en tablas separadas. Es decir una tabla se encuentra en primera forma normal si impide que un atributo de una tupla pueda tomar más de un valor **[70]**.

Poner la base de datos en la Primera Forma Normal resuelve el problema de los encabezados de columnas múltiples.

Todos los grupos repetidos deben dividirse en tablas nuevas, compruebe que la tabla nueva incluya una copia de la llave de la tabla original. La tabla contiene el grupo repetido debe tener una llave concatenada para volver a combinar los datos en las consultas.

La primera forma normal sirve para **[70]**:

- Eliminar los campos repetidos de las tablas individuales
- Crear un grupo de datos

Segunda forma normal (2FN)

Establece que todas las dependientes parciales se deben eliminar y separar dentro de sus propias tablas. Una dependencia parcial es un término que describe a aquellos datos que no dependen de la llave primaria de la tabla para identificarlos. Es decir, cada atributo que no sea clave, depende de forma funcional completa respecto de cualquiera de las claves. La clave principal debe hacer dependientes al

resto de los atributos, si hay atributos que depende sólo de parte de la clave, entonces esa parte de la clave y esos atributos formarán otra tabla.

Cada columna que no es una llave debe depender de la llave para acceder al dato. Solo es un problema con las llaves concatenadas. La solución es dividir las partes que solo dependen de una parte de la llave.

Tercera forma normal (3FN)

Una tabla está normalizada en esta forma si todas las columnas que son llave son funcionalmente dependientes por completo de la llave primaria y no hay dependencias transitivas. Esta ocurre cuando ningún atributo que no sea clave depende transitivamente de las claves de la tabla. Es decir no ocurre cuando algún atributo depende funcionalmente de atributos que no son clave.

La tercera forma normal sirve para [70]:

- Eliminar aquellos campos que no dependan de la clave.
- Relacionar estas tablas mediante una clave externa.

5.5.7.- Sistemas manejadores de bases de datos relacionales

Existe software exclusivamente dedicado a tratar con bases de datos relacionales. Este software se conoce como SGBD (Sistema de Gestión de Base de Datos relacional) o RDBMS (del inglés *Relational Database Management System*) [51].

El *sistema de gestión de la base de datos* (en adelante SGBD) es una aplicación que permite a los usuarios definir, crear y mantener la base de datos, además de proporcionar un acceso controlado a la misma. Se denomina *sistema de bases de datos* al conjunto formado por la base de datos, el SGBD y los programas de aplicación que dan servicio a la empresa u organización [45]. Otro autor comenta

que son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan [43].

El propósito general de los sistemas manejador de bases de datos es el de manejar de manera clara, sencilla y ordenada un conjunto de datos.

Los objetivos de un SMBD son [43]:

- Eliminar la redundancia.
- Mantener la consistencia de los datos.
- Resolver los problemas de concurrencia y regular el acceso a los datos.

Tabla 15.- Ventajas y Desventajas de un SMBD

Ventajas	Desventajas
Provee herramientas que garantizan evitar la duplicidad de registros. Garantiza la integridad referencial, así, al eliminar un registro elimina todos los registros relacionados dependientes. Favorece la normalización por ser más comprensible y aplicable.	Presentan deficiencias con datos gráficos, multimedia, CAD y sistemas de información geográfica. No se manipulan de forma manejable los bloques de texto como tipo de dato. Las bases de datos orientadas a objetos (BDOO) se propusieron con el objetivo de satisfacer las necesidades de las aplicaciones anteriores y así, complementar pero no sustituir a las bases de datos relacionales.

Fuente Elaboración propia con información de [49].

5.5.8.- Definición y funciones de un (DBA)

El administrador de base de datos (DBA) es la persona responsable de los aspectos ambientales de una base de datos. En general sus funciones incluyen las siguientes [43]:

- Recuperabilidad: crear y probar respaldos.
- Integridad: Verificar o ayudar a la verificación en la integridad de datos.
- Seguridad: Definir y/o implementar controles de acceso a los datos.
- Disponibilidad: Asegurarse del mayor tiempo de encendido.

- Desempeño: Asegurarse del máximo desempeño incluso con las limitaciones
- Modificación del esquema y organización física.
- Concesión de la autorización para el acceso a los datos y a la BD.
- Especificación de las limitaciones de consistencia.

5.5.9.- SMDB comerciales y de software libre

De acuerdo con [43] los SMDB comerciales y de software libre son:

SMDB libres:

- Mysql Licencia Dual, depende del uso
- SQLite Licencia Dominio Público
- Apache Derby

SMDB Comerciales:

- Advantage Database
- dBase
- Oracle

5.6.- SQL

5.6.1.- Breve historial SQL

SQL inicio su vida como lenguaje SQUARE en el proyecto del sistema R de IBM. El proyecto del sistema R fue una respuesta de interés generado por el doctor Ted Codd, quien escribió un documento en 1970 sobre base de datos relacionales.

El lenguaje SQUARE fue de alguna manera de naturaleza matemática. Después de conducir experimentos sobre los factores humanos, el equipo de investigaciones de IBM revisó el lenguaje y lo nombro SEQUEL (en lugar SQUARE). Después de otra revisión el lenguaje fue titulado SEQUEL2. Su nombre actual es SQL, resultado de los detalles legales que rodearon el nombre de SEQUEL. Dada esta historia de denominación, existe un número de profesionales de base de datos,

particularmente aquellos que trabajaron durante la década de 1970, que pronuncian el nombre como “sequel” en lugar de SQL (en 1980 se denominó SQL).

En la actualidad SQL es un estándar internacional, aunque no siempre fue así. Con la fuerza de IBM detrás de SQL, muchos imitadores usaron alguna variante de SQL. Así era el antiguo orden en la industria computacional cuando era dominada por IBM. Puede parecernos sorprendente, pero IBM no fue la primera compañía en comercializar SQL. Muchos fabricantes implementaron diferentes subconjuntos de SQL con extensiones propias. El orden se restauró de algún modo gracias a los esfuerzos de estandarización del Instituto Nacional de Estándares de América (ANSI, por sus siglas en ingles), la Organización Internacional de Estándares (ISO) y la comisión electrónica Internacional (IEC). Aunque al inicio SQL no era el mejor lenguaje desarrollado para base de datos, los esfuerzos en los avances de estándares mejorando el lenguaje, así como la estandarización de sus especificaciones [9].

Tabla 16.- Evolución de SQL

Año	Evento
1972	Proyecto del sistema R de los trabajadores de investigación de IBM
1974	Desarrollo del lenguaje SQUARE
1975	Revisión del lenguaje y cambio de nombre a SEQUEL
1976	Revisión del lenguaje y cambio de nombre a SEQUEL 2
1977	Cambio de nombre a SQL
1978	Primera implementación comercial hecha por Oracle Corporation
1981	
1986	IBM produce SQL7DS
1989	Se aprueba el estándar SQL-86 (SQL1)
1992	Se aprueba el estándar SQL-89 (revisión de SQL-86)
1999	Se aprueba el estándar SQL-92 (SQL2)
2003	Se aprueba el estándar SQL:1999 (SQL3)
	Aprobación SQL:2003

Fuente: Elaboración propia con información de [9].

5.6.2.- Lenguaje SQL

- El lenguaje de consulta estructurado (*Structured Query Language*). es el lenguaje estándar ANSI (American National Standards Institute) estándar de la industria, usando para manipular información en una base de datos relacional [67].
- El lenguaje de consulta estructurado es un lenguaje declarativo de acceso a base de datos relacionales que permite especificar diversos tipos de operaciones sobre las mismas [65].
- Es un lenguaje de base de datos normalizado [70].
- Es un lenguaje de consulta estructurado establecido claramente como el lenguaje de alto nivel estándar para sistemas de base de datos relacionales [68].

5.6.3.- Características de SQL

AL respecto [67] menciona las siguientes características de SQL:

- Es un lenguaje basado en el idioma Inglés que usa frases en inglés para manipular la base de datos.
- Es no procedimental. Se especifica la información requerida, más no las operaciones o la forma de llegar a la información. Cada RDBMS tiene un optimizador de búsquedas, el cual traduce sus cláusulas en SQL y elabora el camino o la ruta óptima para llegar a los datos.
- Cuando se trabaja con los datos, todas las filas afectadas por los comandos se tratan como un solo bloque, no son tratadas una por una.
- Adicionalmente, SQL permite que los resultados de una consulta sean los datos de entrada para una nueva.

5.6.4.- Componentes de SQL

El lenguaje SQL está compuesto por comandos, cláusulas, operadores y funciones de agregado. Estos elementos se combinan en las instrucciones para crear, actualizar y manipular las bases de datos [69].

5.6.4.1.- Comandos SQL

Existen dos tipos de comandos SQL:

1. Los comandos DDL (Data Definition Language, Lenguaje de definición de datos) son un subconjunto de comandos SQL que permite hacer modificaciones de la estructura de la base de datos como también para definir derechos de acceso a la base de datos [64].

Con las instrucciones DDL se pueden:

- Definir y crear nuevas tablas.
- Eliminar tablas que ya no sean necesarios.
- Modificar la definición de tablas ya existentes.
- Definir tablas virtuales (o vistas) de los datos.
- Establecer controles de seguridad para la base de datos.
- Crear índices para hacer más rápido el acceso a las tablas.
- Controlar el almacenamiento físico de los datos por el SGBD.

Tabla 17.- Comandos DDL

Comandos DDL	
Comando	Descripción
CREATE	Utilizado para crear nuevas tablas, campos, base de datos e índices. Sintaxis: CREATE DATABASE Nombre BD CREATE TABLE Nombre Tabla (Campo Tipo de dato, Campo Tipo de dato.

DROP	<p>Empleado para eliminar tablas, base de datos e índices.</p> <p>Sintaxis:</p> <p style="text-align: center;">DROP DATABASE Nombre BD DROP TABLE Nombre Tabla</p>
ALTER	Utilizado para modificar las tablas agregando campos o cambiando la definición de los campos.

Fuente: Elaboración propia con información de [70].

2. Lenguaje de manipulación de datos que permiten, DML, es un lenguaje proporcionado por el sistema de gestión de base de datos que permita a los usuarios del mismo llevar a cabo las tareas de consultas o manipulación de los datos, organizados por el modelo de datos adecuados.

Tabla 17.- Comandos DML

Comandos DML	
Comando	Descripción
SELECT	<p>Utilizado para consultar registros de la base de datos que satisfagan un criterio determinado.</p> <p>Sintaxis:</p> <p style="text-align: center;">CREATE DATABASE Nombre BD</p> <p style="text-align: center;">CREATE TABLE Nombre Tabla (Campo Tipo de dato, Campo Tipo de dato.</p>
INSERT	<p>Utilizado para cargar lotes de datos en la base de datos en una única operación.</p> <p>Sintaxis:</p> <p style="text-align: center;">INSERT INTO Nombre de la tabla VALUES (valor1, valor2)</p>
UPDATE	<p>Utilizado para modificar los valores de los campos y registros especificados.</p> <p>Sintaxis:</p> <p style="text-align: center;">UPDATE Nombre de Tabla</p>

	<p>SET Nombre de la columna = criterio</p> <p><</p> <p>></p> <p><></p> <p>WHERE Nombre de columna = criterio</p> <p><</p> <p>></p> <p><></p>
DELETE	<p>Utilizado para eliminar registros de una tabla de una base de datos.</p> <p>Sintaxis:</p> <p>DELETE FROM Nombre de tabla</p> <p>WHERE Nombre de columna = criterio</p> <p><</p> <p>></p> <p><></p>

Fuente: Elaboración propia con información de [70].

5.6.4.2.- Cláusulas

Las cláusulas son condiciones de modificación utilizadas para definir los datos que desea seleccionar o manipular.

Tabla 18.- Cláusulas SQL

Clausulas	Descripción
FROM	Utilizada para especificar la tabla de la cual se van a seleccionar los registros.
WHERE	Utilizada para especificar las condiciones que deben reunir los registros que se van a seleccionar.
GROUP BY	Utilizada para separar los registros seleccionados en grupos específicos.
HAVING	Utilizada para expresar la condición que debe satisfacer cada grupo.
ORDER BY	Utilizada para ordenar los registros seleccionados de acuerdo con un orden específico.

Fuente: Elaboración propia con información de [70].

5.6.4.3- Operadores y condiciones

Los operadores y las condiciones son usados para ejecutar operaciones, tales como la suma y la resta, o comparaciones entre los datos a través de una sentencia de SQL. Los operadores son representados por caracteres simples o por palabras reservadas.

Una condición es una expresión de varios operadores que al ser evaluada retornará VERDADERO, FALSO o DESCONOCIDO.

Los operadores y las condiciones son necesarios en cualquier lenguaje de computación. Ellos te permiten ejecutar cálculos aritméticos, comparaciones de datos, y una amplia variedad de manipulaciones de datos que son necesarios para soportar los requerimientos de las aplicaciones [67].

a) Operadores Lógicos

Los operadores lógicos son usados para producir un resultado simple a partir de la combinación de dos condiciones por separado [67].

Tabla 19.- Operadores lógicos SQL

Operador	Uso
AND	Es el "y" lógico. Devuelve verdadero si las expresiones a su izquierda y derecha son ambas Verdaderas.
OR	Es el "o" lógico. Devuelve verdadero si cualquiera de las dos expresiones a izquierda y derecha del OR, son verdaderas.
NOT	Negación lógica. Invierte la lógica de la expresión que está a su derecha. Si era verdadera, mediante NOT pasa a ser falso.

Fuente: Elaboración propia con información de [70].

b) Operadores de Comparación

Los operadores de comparación son usados para comparar una expresión contra otra expresión. El resultado de la comparación debe conducir a un valor Falso, Verdadero o Desconocido [67].

Tabla 20.- Operadores de comparación SQL

Operador	Uso
<	Menor que.
>	Mayor que.
<>	Distinto de.
<=	Menor ó Igual que.
>=	Mayor ó Igual que.
=	Igual que.
BETWEEN	Utilizado para especificar un intervalo de valores.
LIKE	Utilizado en la comparación de un modelo.
IN	Utilizado para especificar registros de una base de datos.

Fuente: Elaboración propia con información de [70].

c) Operadores de caracteres

Las cadenas de caracteres son manipuladas por los operadores de carácter. El más común de los operadores de carácter es el operador de concatenación.

Tabla 21.- Operadores de caracteres SQL

Operador	Uso
	Concatena cadenas de caracteres.

Fuente: Elaboración propia con información de [67].

El resultado de concatenar dos cadenas de carácter es otra cadena de carácter. Si ambas cadenas son del mismo tipo de datos, la cadena resultante será del mismo tipo de datos original [67].

d) Operadores matemáticos

Los operadores aritméticos son usados en SQL para sumar, restar, multiplicar, dividir y negar valores. El resultado de una expresión que use operadores aritméticos es un valor numérico [67].

Tabla 22.- Operadores matemáticos SQL

Operador	Uso
+	Suma
-	Utilizada para devolver el número de registros de la selección
*	Utilizada para devolver la suma de todos los valores de un campo determinado
/	Utilizada para devolver el valor más alto de un campo especificado
^	Utilizada para devolver el valor más bajo de un campo especificado

Fuente: Elaboración propia con información de [67].

e) Alias

En determinadas circunstancias es necesario asignar un nombre a alguna columna determinada de un conjunto devuelto. Para resolver este problema tenemos la palabra reserva AS que se encarga de asignar el nombre que deseamos a la columna deseada.

Ejemplo:

```
SELECT importe * cantidad AS total
FROM Pedidos
```

5.6.4.4.- Funciones de agregado

a) Funciones de Agregado

SQL permite resumir los datos de la base de datos con un conjunto de funciones de columna. Una función de columna toma los datos de ésta como rango de valores y produce un único resultado.

Sintaxis:

```
SELECT Nombre función (Nombre Consulta)
FROM Nombre Tabla
```

Tabla 23.- Funciones de agregados SQL

Función	Descripción
AVG	Calcula la media aritmética de un conjunto de valores contenidos en un campo especificado de una consulta. SELECT SUM (Sueldo) FROM tbl_empleados
COUNT	Cuenta los elementos de un grupo. Se utiliza el asterisco para no tener que indicar un nombre de columna concreto, el resultado es el mismo para cualquier columna.
SUM	Devuelve la suma del conjunto de valores contenido en un campo específico de una consulta.
MAX, MIN	Devuelven el mínimo o el máximo de un conjunto de valores contenidos en un campo específico de una consulta.
VAR, VARP	Devuelve una estimación de la varianza de una población (sobre el total de los registros) o una muestra de la población (muestra aleatoria de registros) sobre los valores de un campo.
STDEV, STDEVP	Devuelve estimaciones de la desviación estándar para la población (el total de los registros de la tabla) o una muestra de la población representada (muestra aleatoria)

Fuente: Elaboración propia con información de [69].

b) Funciones de caracteres

Conversión del texto a mayúsculas y minúsculas Son:

Tabla 24.- Funciones de caracteres SQL

Función	Uso
LOWER(<i>texto</i>)	Convierte el texto a minúsculas (funciona con los caracteres españoles)

UPPER(<i>texto</i>)	Convierte el texto a mayúsculas
INITCAP(<i>texto</i>)	Coloca la primera letra de cada palabra en mayúsculas

Fuente: Elaboración propia con información de[45]

5.6.5.- Consulta

La consulta es el método para acceder a los datos, por medio de ella se puede agregar, ordenar, procesar, filtrar, manejar y modificar en cualquier forma los datos que se tiene en las tablas de la base de datos.

De manera que podría decirse que las consultas tienen como función la creación de la información para la buena toma de decisiones, ya que mediante ellas se transforman los datos que se tienen para crear un conjunto de significación relevante dentro de un área [65].

- **Consultas simples**

Las consultas simples son aquellas que obtienen información de una única tabla. La base de la consulta es la cláusula SELECT [70].

La forma simple de una consulta es a través de la siguiente sintaxis:

```
SELECT Nombre de columna
FROM Nombre Tabla
```

1.- Nombre de columna. Especifica las columnas con las alias o, en el caso de que sean todas, el símbolo (*) se podrá utilizar. Si los alias contienen espacios en blanco, deben de estar entre corchetes.

2.- Nombre de la tabla. La tabla es el origen de donde se obtendrán los datos. Sin la tabla contiene espacios en blancos, deben estar entre corchetes.

Ejemplo:

```
SELECT Empleado, RFC, CURP
FROM Tbl_empleado
```

- **Consultas con Predicado**

El predicado se incluye entre la cláusula y el primer nombre del campo a recuperar, los posibles predicados son:

Tabla 25.- Consultas con predicados

ALL	Devuelve todos los campos de la tabla <i>SELECT ALL FROM Empleados</i>
TOP	Devuelve un cierto número de registros que entran entre al principio o al final de un rango especificado de la tabla. <i>SELECT TOP 25 Nombre, Apellido FROM Estudiantes ORDER BY Nota DESC</i>
DISTINCT	Omite los registros cuyos campos seleccionados coincidan totalmente. <i>SELECT TOP 10 PERCENT Nombre, Apellido, Materno FROM tbl_empleado ORDER BY Nota DESC</i>
DISTINCTROW	Omite los registros duplicados basándose en la totalidad del registro y no sólo en los campos seleccionados. <i>SELECT DISTINCT Apellido FROM Empleados</i>
*	Devuelve todos los campos de la tabla. <i>SELECT * FROM Empleados</i>

Fuente: Elaboración propia con información de [45].

5.6.6.- Cláusula WHERE

La cláusula WHERE puede usarse para determinar qué registros de las tablas enumeradas en la cláusula FROM aparecerán en los resultados de la instrucción SELECT. Si no se emplea esta cláusula, la consulta devolverá todas las filas de la tabla. WHERE es opcional, pero cuando aparece debe ir a continuación de FROM [66].

La forma simple de una consulta con WHERE es a través de la siguiente sintaxis [70]:

```
SELECT Nombre de columna  
FROM Nombre tabla  
WHERE Condición
```

- 1.- Nombre de columna. Especifica las columnas con las alias o, en el caso de que sean todas, el símbolo (*) se podrá utilizar. Si los alias contienen espacios en blanco, deben de estar entre corchetes.
- 2.- Nombre de la tabla. La tabla es el origen de donde se obtendrán los datos. Sin la tabla contiene espacios en blancos, deben estar entre corchetes.
- 3.- WHERE (Condición). Esta cláusula consiste en una condición de búsqueda que especifica las filas a recuperar, si la condición es un valor alfanumérico se requiere de una comilla sencilla (') si es un valor numérico no la requiere.

La condición está acompañada por los siguientes operadores =, <>, <, >, <=, >=.

Ejemplo

```
SELECT Empleado, RFC, CURP  
FROM Tbl_empleado  
WHERE n_empleado = 123
```

5.6.7.- Cláusula ORDER BY

Se puede pedir que SQL, ordene los resultados incluyendo la cláusula ORDER BY en la instrucción SELECT. La cláusula ORDER by va seguida de una lista de especificaciones de orden separadas por comas.

La forma simple de una consulta ORDER BY es a través de la siguiente sintaxis [70]:

```
SELECT Nombre de columna
FROM Nombre tabla
ORDER BY campo, campo ASC
DESC
```

- 1.- Nombre de columna. Especifica las columnas con las alias o, en el caso de que sean todas, el símbolo (*) se podrá utilizar. Si los alias contienen espacios en blanco, deben de estar entre corchetes.
- 2.- Nombre de la tabla. La tabla es el origen de donde se obtendrán los datos. Sin la tabla contiene espacios en blancos, deben estar entre corchetes.
- 3.- ORDER BY Solicita el orden en secuencia Ascendente o Descendente de la columna asignada y se puede organizar por cualquier elemento de la lista.

Ejemplo:

```
SELECT Empleado, RFC, CURP
FROM Tbl_empleado
ORDER BY n_empleado DESC
```

5.6.8.- Cláusula GROUP BY

Se usa para generar valores de agregado para cada fila del conjunto de resultados [70].

```
SELECT Nombre función, Nombre de columna  
FROM Nombre tabla  
WHERE condición  
GROUP BY Nombre columna
```

5.6.9.- Cláusula HAVING

Se puede usar para seleccionar y descartar los grupos de filas. El formato de la cláusula HAVING es paralelo al de la cláusula WHERE que consiste en la palabra clave HAVING seguida de una condición de búsqueda. La cláusula HAVING especifica así una condición de búsqueda para grupo. HAVING se coloca generalmente cerca del fin de la instrucción SQL [70].

```
SELECT Nombre función FUNCION (Nombre  
de columna)  
FROM Nombre tabla  
GRUP BY Nombre columna  
HAVING condición
```

VI. METODOLOGÍA

6.1.- Pasos Metodológicos

El objetivo de esta metodología es la definición de las tareas que se realizaron, el producto a obtener y las técnicas que empleamos. Para ello siguieron los siguientes pasos:

Paso 1.- Estructuración de la entrevista para conocer las necesidades del sistema y delimitar el alcance del sistema de información.

Paso 2.- Determinar las reglas de negocios que reflejen las necesidades de los usuarios.

Paso 3.- Identificar datos y reportes.

Paso 4.- Diseño del modelo relacional de la base de datos.

Paso 5.- Implementación física de la base de datos en MySQL.

Paso 6.- Análisis y diseño del sistema de información en UML, casos de uso.

Paso 7.- Diseño de vistas del sistema.

Paso 8.- Codificar o Programar los algoritmos previamente diseñados.

Paso 9.- Implementación del Sistema de Información.

Paso 10.- Validación y Pruebas del Sistema de Información.

6.1.1.- Estructuración de la entrevista para conocer las necesidades del sistema y delimitar el alcance del sistema de información

Lo primero fue realizar una entrevista (se utilizó la entrevista abierta porque posibilita una investigación más amplia y profunda de la personalidad del entrevistado.) para conocer el funcionamiento de la Secretaria de Administración Tributaria con el propósito de mejorar los procesos de la misma y se documentaron los resultados de dicha entrevista en el Anexo1.

6.1.2.- Determinar las reglas de negocios que reflejen las necesidades de los usuarios.

Todo surgió a partir de la problemática que se tiene en una de las Administraciones Generales perteneciente a la Secretaría de Administración Tributaria (SAT) en el Departamento de Recursos Humanos de dicha administración.

Se identificaron los requerimientos del Departamento de Recursos Humanos del SAT, para llevar el control de empleados, los cuales fueron los siguientes:

- Reducir la pérdida de tiempo en buscar la información requerida.
- La entrega de reportes quincenales, mensuales y anuales en tiempo y forma.
- Asegurar la autenticidad del historial de cada empleado.
- Mantener actualizada la estructura de cada área con respecto al personal contratado.
- Generar información en tiempo real.
- Realizar el pago de nómina sin tener ningún atraso por contar con datos en tiempo real.
- Contar con un historial adecuado del personal para los reingresos no deseados.
- Toma de decisiones al contar con información en tiempo real.

Se determinó el propósito que el SAT trata de conseguir al implementar un sistema de información en el Departamento de Recursos Humanos, como sigue:

- Determinar las necesidades del sistema de información para delimitar el alcance del trabajo planteado.
- Establecer las reglas de negocios del sistema de información para cumplir con los requerimientos del usuario.
- Diseñar el modelo de bases de datos para integrar el sistema de información.
- Planteamiento de los algoritmos del sistema de información para cumplir con las reglas de negocios.
- Programación de los algoritmos para implementar el sistema de información.
- Estructuración del sistema de información para su operatividad.

6.1.3.- Identificar datos y reportes

Se visualizaron las especificaciones del sistema a elaborar, utilizando un Diagrama de flujo de datos (DFD) que es un modelo que describe los flujos de datos, los procesos que cambian o transforman los datos en un sistema, las entidades externas que son fuente o destino de los datos y los almacenamientos de datos a los cuales tiene acceso el sistema, permitiendo así describir el movimiento de los datos a través del sistema.

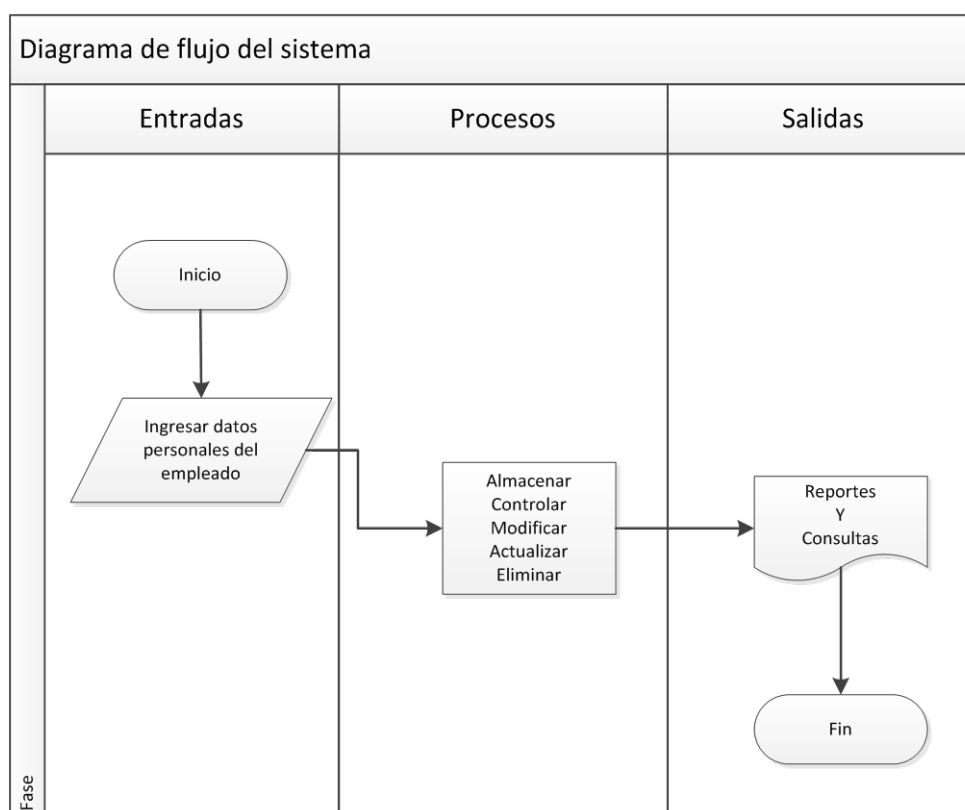


Figura 17 Diagrama de flujo del sistema de Recursos Humanos. Fuente: Elaboración propia, 2013.

Utilizando el diagrama de flujo anterior para graficar las entradas, los procesos y las salidas de las funciones del departamento. A partir de los diagramas de flujo se desarrolló un diccionario de datos que es un listado organizado de todos los datos que pertenecen al sistema. El objetivo de éste, es dar precisión sobre los datos que se manejan en un sistema, evitando así malas interpretaciones o ambigüedades.

A continuación se describe de manera breve características individuales de cada dato que conforman la base de datos, así como sus respectivas especificaciones para identificar los datos necesarios para la automatización de procesos.

Tabla 26.- Diccionario de Datos.

Tabla_Empleado (tbl_empleado)					
CAMPO	MySQL	DESCRIPCIÓN	TIPO DE DATO	TAMAÑO	DESCRIPCION DE LA TABLA
Numero de empleado	n_empleado	Es el número con en el que se identifica el empleado, este es asignado desde que entra a laborar	Numérico	10	En esta tabla se registran los datos personales de los empleados.
Nombre del empleado	Nombre	Nombre del empleado	Carácter	100	
Paterno	Paterno	Apellido paterno del empleado	Carácter	100	
materno	materno	Apellido materno del empleado	Carácter	100	
RFC	rfc	Registro federal de causantes	Carácter	15	
RFC corto	rfc_corto	Registro federal de causantes pero solo los primeros 6 dígitos	Carácter	10	
CURP	Curp	Clave única de registro conformada con los datos personales del empleado.	Carácter	18	

Tabla_Banco (tbl_banco)					
CAMPO	MySQL	DESCRIPCIÓN	TIPO DE DATO	TAMAÑO	DESCRIPCION DE LA TABLA
Id banco	Id_banco	Identificador de la tabla banco	Numérico	11	En esta Tabla se registran los datos bancarios de cada empleado.
Cuenta bancaria	cuenta_ban	Clave que se le asigna al empleado para deposito de sueldo.	Numérico	40	
Clave interbancaria	clave_inter	Clave personal que el banco le asigna al Empleado.	Numérico	40	
Banco	banco	Nombre del banco al que está inscrito.	Carácter	25	
Número de empleado	n_empleado	Número del empleado	Numérico	10	

Tabla_Bajas (tbl_bajas)					
CAMPO	MySQL	DESCRIPCIÓN	TIPO DE DATO	TAMAÑO	DESCRIPCION DE LA TABLA
Id de la baja	Id_baja	Identificador de la tabla bajas	Numérico	4	En esta tabla se registraran los datos referentes a la baja de un empleado
Id de la causa baja	Id_cbaja	Identificador de la tabla Causa baja	Numérico	4	
Numero de contrato	n_contrato	Clave conformada por números que identifican determinado contrato.	Carácter	20	
Atenta Nota	a_notas	Número que identifican con que atenta nota se tramita la baja .	Numérico	11	

Tabla Plaza (tbl_plaza)					
CAMPO	MySQL	DESCRIPCIÓN	TIPO DE DATO	TAMAÑO	DESCRIPCIÓN DE LA TABLA
Número posición	n_posicion	Número de 8 dígitos que identifica la plaza.	Numérico	11	En esta tabla se registran los datos de las características de las plazas del departamento
Estado	estado	Se refiere al estado de la plaza si esta Ocupada, Vacante, Comisionada, Vacante congelada.	Carácter	20	
Régimen	regimen	Eventual o presupuestal.	Carácter	12	
Número Departamento	n_dep	Es el número del departamento que identifica la posición que tiene el empleado en el organigrama de su administración.	Carácter	11	

Tabla_Historial (tbl_historial)					
CAMPO	MySQL	DESCRIPCIÓN	TIPO DE DATO	TAMAÑO	DESCRIPCIÓN DE LA TABLA
Id de Historial	Id_historial	Identificador de la tabla historial.	Numérico	11	En esta tabla se concentraran los datos los movimientos de un empleado durante su instancia en la organización.
Id de movimiento	Id_mov	Identificador de la tabla movimiento	Numérico	11	
Fecha de inicio	f_inicio	Fecha inicial del contrato	Fecha		
Fecha de terminación	f_termino	Fecha de término del contrato	Fecha		
Número de posición	n_posicion	Numero de 8 dígitos que identifica la plaza.	Numérico	10	
Número del empleado	n_empleado	Es el numero con en el que se identifica el empleado, este es asignado desde que entra a laborar.	Numérico	11	
Id de área	Id_area	Identificador de la tabla área.	Numérico		
Código número	codigo_n	Es el nivel salarial del empleado	Carácter	6	
Número de contrato	n_contrato	Clave conformada por números que identifican determinado contrato.	Carácter	20	
Id de ubicación	Id_ubicacion	Identificador de la tabla ubicación.	Numérico	11	
Id empleado del jefe	Id_empleado_jefe	Identificador del nombre del empleado	Numérico	11	

Catálogo_Nivel (cta_nivel)					
CAMPO	MySQL	DESCRIPCIÓN	TIPO DE DATO	TAMAÑO	DESCRIPCIÓN DE LA TABLA
Código del nivel	codigo_n	Clave de letras y números que identifica el nivel salarial	Carácter	8	En este catálogo se almacena los códigos de los niveles existentes en el Departamento.
Nivel	nivel	Es el nivel salarial del empleado	Carácter	5	
Sueldo del empleado	sueldo	Cantidad mensual que se le pagará al empleado	Numérico	11	
Descripción	descrip	Es el cargo correspondiente al nivel (Administrador, Jefe de departamento...)	Carácter	100	
Jerarquía	jerarquia	Define la jerarquía del nivel	Carácter	1	

Catálogo_Área (cta_area)					
CAMPO	MySQL	DESCRIPCIÓN	TIPO DE DATO	TAMAÑO	DESCRIPCIÓN DE LA TABLA
Id de área	id_area	Identificador de la tabla área	Numérico	11	En este catálogo se almacena los códigos de las áreas existentes en el Departamento
Nombre del área	n_area	Nombre de la administración	Carácter	100	
Letra del área	letra	Es una letra que identifica a la administración.	Carácter	1	

Catálogo_Rango (cta_rango)					
CAMPO	MySQL	DESCRIPCIÓN	TIPO DE DATO	TAMAÑO	DESCRIPCIÓN DE LA TABLA
Código del rango	c_rango	Clave de letras y números que hace referencia al cargo y administración correspondiente.	Carácter	8	En este catálogo se almacena rangos de los niveles establecidos por cada área del Departamento
Nombre del rango	n_rango	Cargo específico del empleado según la administración a la que corresponda.	Carácter	100	
Id área	id_area	Identificador de la tabla área	Numérico	11	
Rango inferior nivel	r_infer_nivel	Nivel mínimo que puede tener un empleado en determinada administración.	Carácter	4	
Rango superior nivel	r_super_n	Nivel máximo que puede tener un empleado en determinada administración.	Carácter	4	

Catálogo_Causa_baja (cta_causa_baja)					
CAMPO	MySQL	DESCRIPCIÓN	TIPO DE DATO	TAMAÑO	DESCRIPCIÓN DE LA TABLA
Id de la causa baja	Id_cbaja	Identificador de la tabla Causa baja	Numérico	11	En este catálogo se almacena los diferentes tipos de bajas que maneja el departamento
Nombre de la causa baja	n_cbaja	Son los tipos de bajas (renuncia, jubilación, pensión...)	Carácter	30	

Catálogo_Movimiento (cta_mov)					
CAMPO	MySQL	DESCRIPCIÓN	TIPO DE DATO	TAMAÑO	DESCRIPCIÓN DE LA TABLA
Id de movimiento	Id_mov	Identificador de la tabla movimiento	Numérico	11	En este catálogo se almacenan los diferentes tipos de movimientos que maneja el departamento
Nombre del movimiento	n_mov	Son los tipos de movimientos (permuta, promoción, nuevo ingreso...).	Carácter	25	

Catálogo_Ubicación (cta_ubicacion)					
CAMPO	MySQL	DESCRIPCIÓN	TIPO DE DATO	TAMAÑO	DESCRIPCIÓN DE LA TABLA
Id de ubicación	Id_ubicacion	Identificador de la tabla ubicación	Numérico	11	En este catálogo se almacenaran las características del lugar físico en donde se encuentran ubicados los empleados
Modulo	modulo	Numero de edificio al que pertenece el empleado.	Carácter	15	
Piso	piso	Número del piso del edificio al que pertenece el empleado.	Numérico	11	
Ubicación	ubicación	Clave de letras y numero que identifica el numero del edificio y el piso al que pertenece el empleado.	Carácter	15	

Fuente: Elaboración propia

Reportes

Los reportes utilizados para el control eficiente del personal del departamento de Recursos Humanos son:

- Reportes de Movimientos de los empleados de manera global.
- Reportes de Movimientos de los empleados por área.
- Reportes de Movimientos de los empleados por periodo determinado.
- Reportes de bajas de los empleados de manera global.
- Reportes de bajas de los empleados por área.
- Reportes de bajas de los empleados por periodo determinado.

- Reporte histórico por empleado.
- Reporte resumen de empleados de manera global.
- Reporte resumen de empleados por tipo de empleado.
- Reporte resumen de empleados por área.
- Reporte resumen de empleados por nivel.

6.1.4.- Diseño del modelo relacional de la base de datos.

Se utilizó el modelo relacional para el diseño de la base de datos para el sistema de Recursos humanos. La cual está integrada por 5 tablas y 7 catálogos

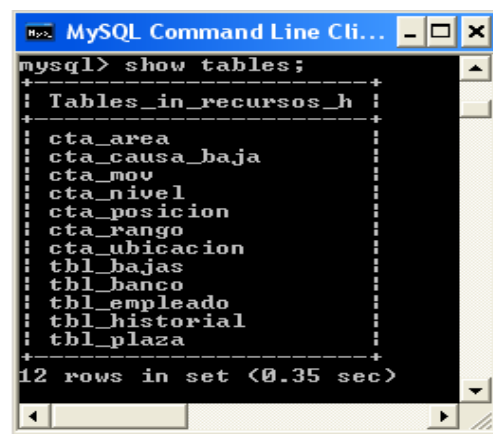


Figura 18 Tablas y catálogos realizados en MySQL. Fuente: Elaboración propia

6.1.5.- Implementación física de la base de datos en MySQL.

La base de datos que almacenará gran parte de los datos indispensables, fue desarrollada en MySQL 5.1 se decidió realizarla en este gestor porque es una plataforma de trabajo bastante sencilla de utilizar, cuenta con una licencia de funcionamiento GPL (Licencia Pública General).

Además de que se adapta a nuestras necesidades ya que permite acceder a todos los campos que resguardan los datos de trabajo, puede ingresar una enorme cantidad de datos por columna de trabajo. Capacidad de soportar hasta 32 índices

de tablas diferentes y su capacidad de realizar pruebas de rendimiento en tiempo real y este gestor cuenta con una gran estabilidad ya que una base de datos bien organizada es el cimiento de cualquier sistema de información.



Figura 19 Mysql Fuente MySQL. Fuente: [73]

6.1.6.- Análisis y diseño del sistema de información en UML, casos de uso.

Los procesos de sistema se elaboraron utilizando diagramas de UML, a través de los de casos de uso, que dan una mejor visualización en las actividades a realizar en cada proceso.

Casos de uso para dar de alta a un Empleado.

El sistema permite dar de alta a los nuevos empleados indefinidamente en la base de datos. En la figura 20 se visualiza este proceso.

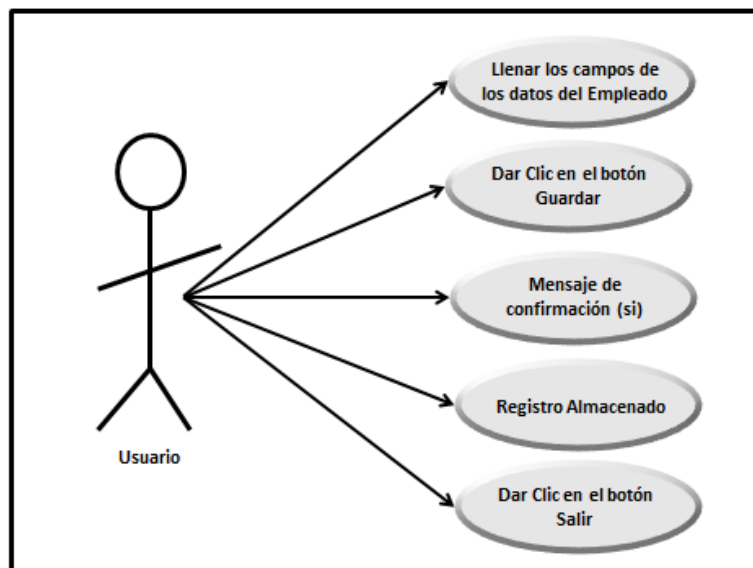


Figura 20 Casos de uso Guardar. Fuente: Elaboración propia, 2013.

Casos de usos para buscar un Empleado

El sistema permite buscar a cualquier empleado almacenado en la base de datos y muestra el estatus dentro de la administración. Como se muestra en la figura 21:

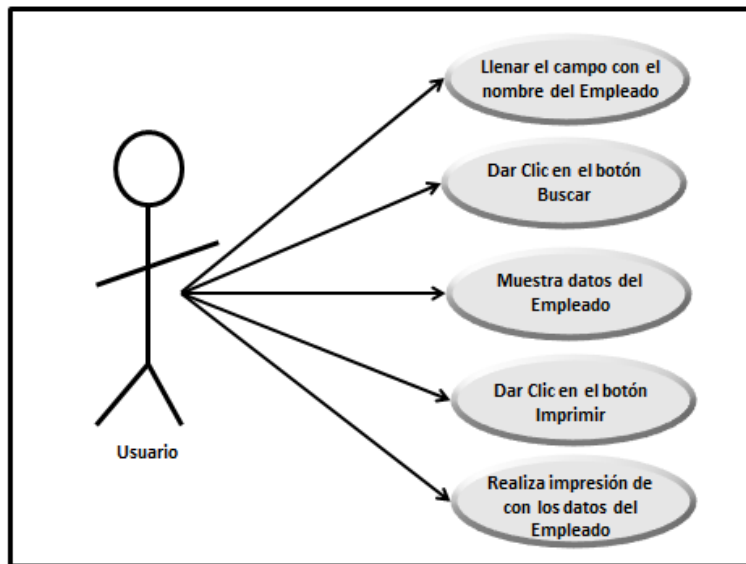


Figura 21 Casos de uso Buscar. Fuente: Elaboración propia, 2013.

Casos de uso para modificar datos de un Empleado.

El sistema da la opción de modificar cualquier dato relacionado con el empleado y guardar la modificación permitiendo así mantener actualizada la base de datos. Como se observa en la figura 22:

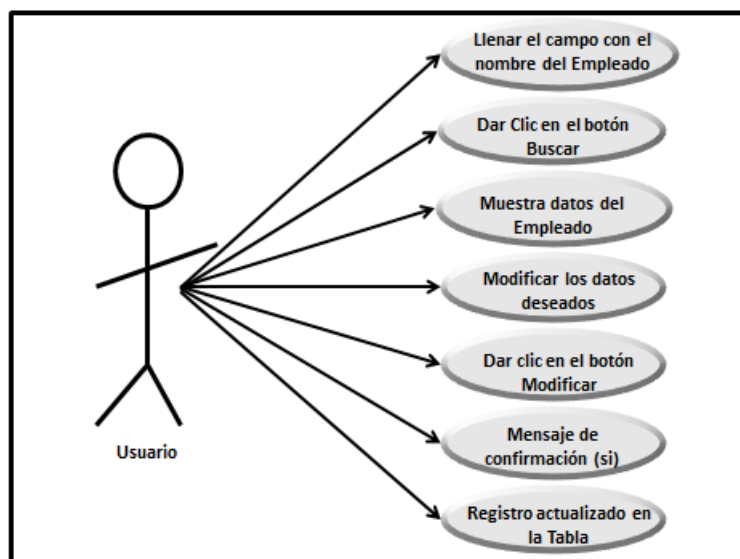


Figura 22 Casos de uso Modificar. Fuente: Elaboración propia, 2013.

Casos de uso para generar reportes.

La herramienta principal de este sistema es la generación de reportes múltiples que permite realizar en tiempo real de acuerdo a los parámetros seleccionados por el usuario e imprimirlos. Como se muestra en la figura 23:

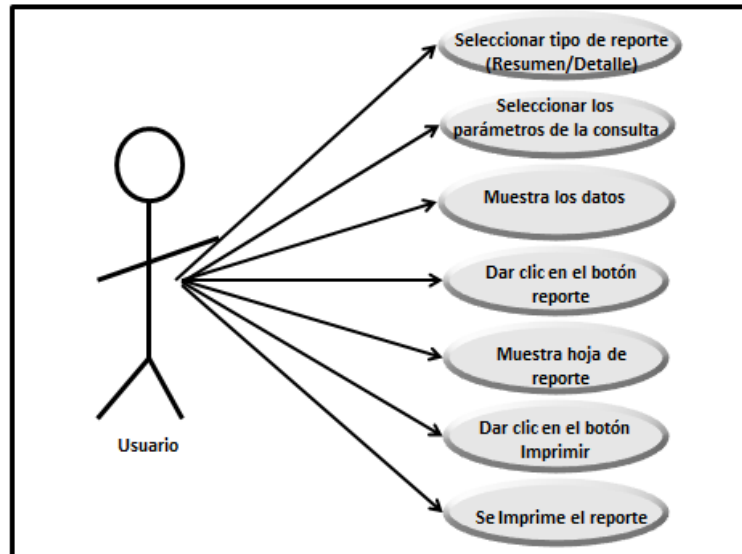


Figura 23 Casos de uso Generación de Reportes. Fuente: Elaboración propia, 2013.

Casos de uso exportación de reportes

El sistema imprime los reportes a Excel mediante la opción de exportar, como otra alternativa de llevar el control de la administración. Como se muestra en la figura 24.

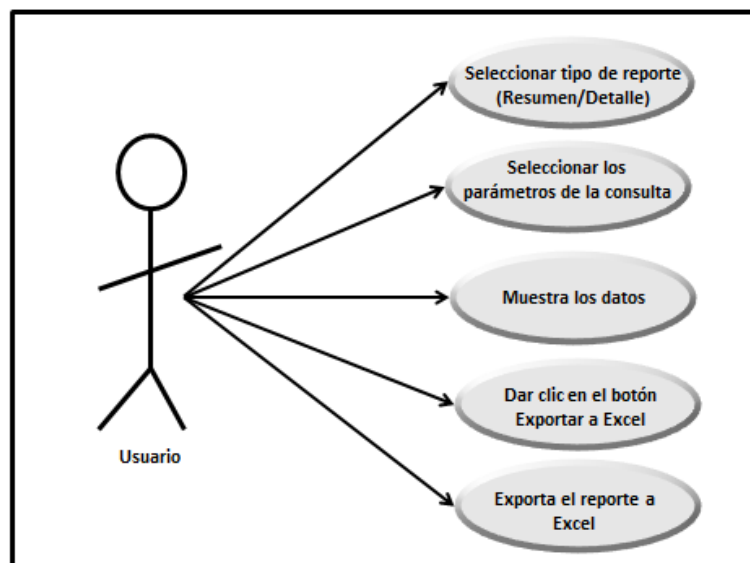


Figura 24 Casos de uso exportación de reportes. Fuente: Elaboración propia, 2013.

6.1.7.- Diseño de vistas del sistema.

Para el diseño de las vistas del sistema se tomó en cuenta las necesidades de los usuarios. Se desarrollaron bocetos de las posibles interfaces que fueran sencillas, amigables para el usuario, capaces de satisfacer las necesidades identificadas.



Figura 25 Vista principal del sistema en Visual Basic 6.0. Fuente: Elaboración propia, 2013.

6.1.8.- Codificar o Programar los algoritmos previamente diseñados.

Para la codificación de los algoritmos que darán solución a las necesidades del departamento de recursos humanos se utilizó un lenguaje de programación Visual, ya que es un lenguaje intuitivo, Se desarrolla bajo un entorno totalmente gráfico lo que hace que su manejo se haga más ameno.

Permite crear aplicaciones para Windows en muy poco tiempo. En otras palabras, permite un desarrollo eficaz y menor inversión en tiempo que con otros lenguajes.

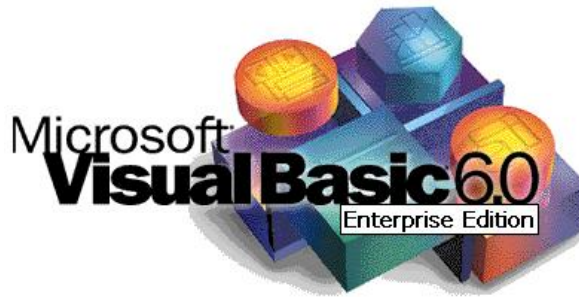


Figura 26 Visual Basic 6.0. Fuente: Visual Basic 6.0. Fuente: [74]

6.1.9.- Implementación del Sistema de Información

Después de analizar los modelos de procesos de software se decide utilizar el modelo en espiral ya que a diferencia de otros modelos de proceso que finalizan cuando se entrega el software, el modelo en espiral puede adaptarse y aplicarse a lo largo de la vida útil. Por lo tanto, el primer círculo o vuelta alrededor del espiral podría representar un “proyecto de desarrollo del concepto”, el cual se inicia en su centro y continúa por múltiples iteraciones hasta completar el desarrollo [52].

Además [54], argumenta que la diferencia principal entre el modelo espiral y los modelos anteriores (ejemplo: cascada, evolutivo, incremental.) es la evaluación del riesgo. El riesgo es todo lo que pueda salir mal en un proyecto de desarrollo de software. Por ejemplo, si queremos utilizar un lenguaje de programación para desarrollar un sistema operativo, un riesgo posible es que los compiladores utilizables no produzcan un código objeto eficiente. Los riesgos originan problemas en el proyecto, como el exceso de los costos. Es así que, la disminución de los riesgos es una actividad muy importante.

Para el desarrollo del sistema de recursos humanos es necesario crear formularios que permitan mantener la estructura de dicho sistema, por este motivo se crearon formularios tales como se muestran en la figura 27.

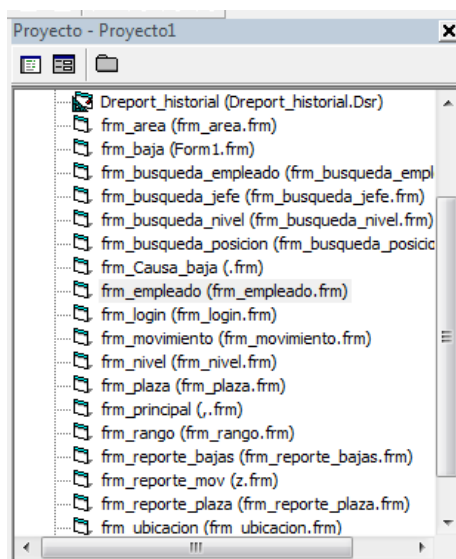


Figura 27 formularios del sistema. Fuente: Elaboración propia, 2013.

Los reportes se presentaron en el formato requerido por el usuario, para la administración del capital humano, se realizaron con sentencias SQL, un ejemplo de la forma en que se presentan se puede ver en la Figura 28.

Reporte de movimientos

Zoom 100%

Administración General de Grandes Contribuyentes
Subadministración de Enlace Administrativo
Departamento de Recursos Humanos

NOMBRE DEL ÁREA	N	NOMBRE	POSICIÓN	MOVIMIENTO	NIVEL	COD PUESTO	REGIMEN	F. DE INICIO
ADMINISTRACION CENTRAL DE PLANEACION Y PROGRAMACION	92722	JUAN JOSE ROMERO LOPEZ	1030098	NUEVO INGRESO	KA03	GC2012	Presupuestal	11/01/2013
ADMINISTRACION CENTRAL DE PLANEACION Y PROGRAMACION	85711	SONIA BELMON SALAZAR	10321031	NUEVO INGRESO	PQ2	GC6094	Presupuestal	11/01/2013
ADMINISTRACION CENTRAL DE PLANEACION Y PROGRAMACION	77810	SANDRA VELAZQUEZ SOLAR	10330649	NUEVO INGRESO	SB01	GC4080	Presupuestal	16/02/2013
ADMINISTRACION CENTRAL DE PLANEACION Y PROGRAMACION	65432	EDUARDO RUIZ MORENO	10341899	NUEVO INGRESO	DA11	GC5081	Presupuestal	16/02/2013
ADMINISTRACION CENTRAL DE PLANEACION Y PROGRAMACION	112532	MARIA LARA PEREZ	20200050	NUEVO INGRESO	PC3	GC6094	eventual	01/01/2013
ADMINISTRACION CENTRAL DE NORMATIVIDAD	52781	FABIOLA ZAVALA CRUZ	10300010	NUEVO INGRESO	KA06	GC2010	Presupuestal	01/03/2013
ADMINISTRACION CENTRAL DE NORMATIVIDAD	89121	FERNANDA CARRILLO PADILLA	10300479	NUEVO INGRESO	11	GC6071	Presupuestal	01/02/2013
ADMINISTRACION CENTRAL DE NORMATIVIDAD	10001	ROBERTO RAMIREZ ZORRILLA	10349163	NUEVO INGRESO	SC08	GC4072	Presupuestal	16/03/2013
ADMINISTRACION CENTRAL DE NORMATIVIDAD	32574	OSCAR SOLANO BARRERA	10324807	NUEVO INGRESO	DC10	GC5073	Presupuestal	16/02/2013
ADMINISTRACION CENTRAL DE NORMATIVIDAD	69320	ALFREDO JUAREZ OTANES	10324814	NUEVO INGRESO	PQ3	GC6071	Presupuestal	16/03/2013
Administración Central de dineros	92779	STEPHANIE CASTILLO CAMACHO	10300618	NUEVO INGRESO	KA20	GC2003	Presupuestal	16/02/2013
Administración Central de dineros	8534	JOSEFINA MARTINEZ GOMEZ	20200263	NUEVO INGRESO	PQ1	GC6043	eventual	01/01/2013
Administración Central de dineros	12564	MARIANA RAMOS LIMA	20200817	NUEVO INGRESO	SB09	GC4039	eventual	16/01/2013
Administración Central de dineros	69102	ROSA MUÑOZ RODRIGUEZ	10331671	NUEVO INGRESO	DA08	GC5040	Presupuestal	01/02/2013

Páginas: 1

Figura 28 Reporte del sistema en Visual Basic 6.0. Fuente: Elaboración propia, 2013.

6.1.10.- Validación y Pruebas del Sistema de Información

Para la validación del sistema se requirió que los encargados del departamento de recursos humanos llevaran a cabo pruebas con los catálogos llenados de acuerdo los criterios y lineamientos del departamento para verificar que la información que mostrará el sistema fuera correcta actual acorde a la valuación de los puestos asignados a cada uno de los empleados.

Para almacenar los datos se diseñó la base de datos relacional que se muestra en la Figura 29. Con los datos contenidos en ella se realizaron las consultas SQL.

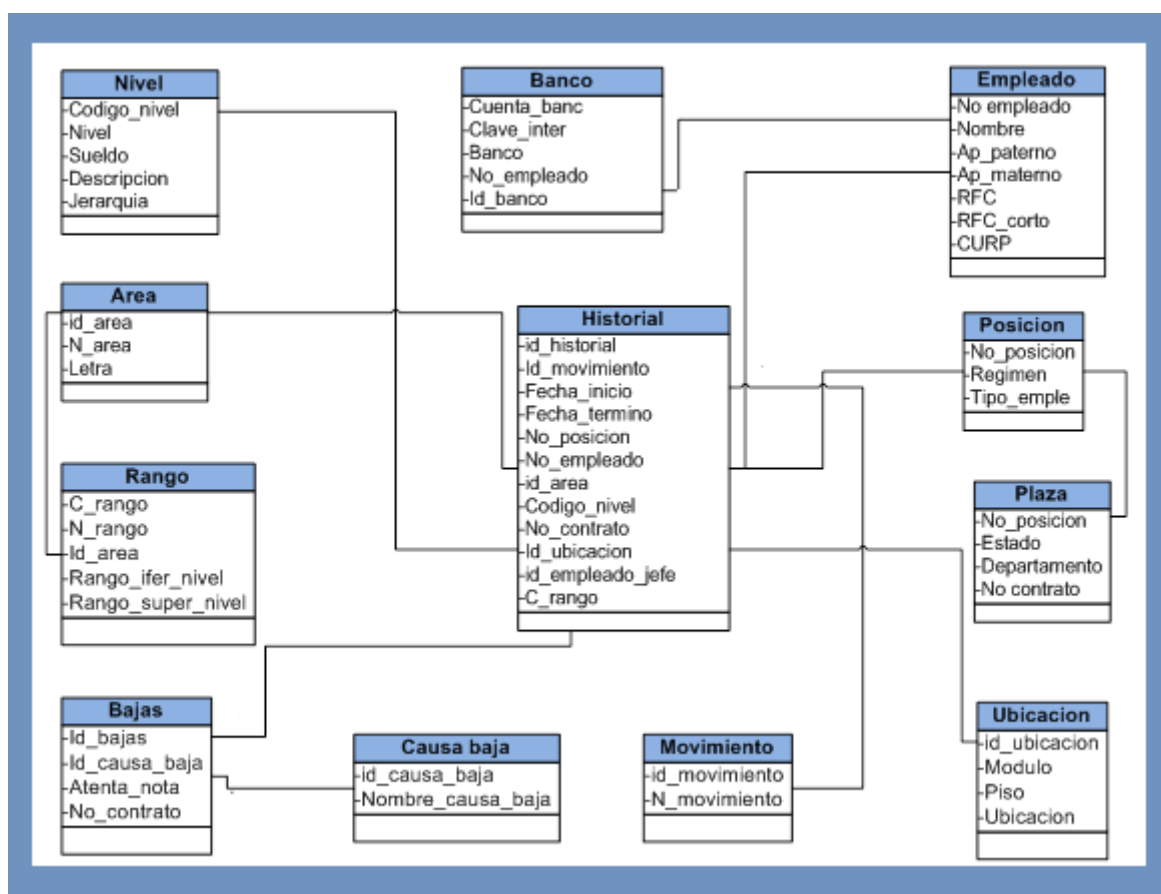


Figura 29 Base de Datos Relacional del sistema para administración de Recursos Humano
Fuente: Elaboración propia, 2013.

VII.- RESULTADOS

Como resultado de esta investigación se desarrolló un sistema diseñado para satisfacer las necesidades del departamento de recursos humanos de la Administración General del Servicio de Administración Tributaria (SAT-System), teniendo los beneficios de que se mencionan en el tabla 27.

Tabla 27.- Sistema de información de recursos humanos SAT System.

Nombre	Beneficios
NeFab System	<ul style="list-style-type: none">• Mantener en todo momento un expediente completo del personal.• Muestra información de cada empleado a través de una sola vista y haciéndolo más sencillo.• Consultar y administrar al personal de la empresa.• Consulta de historial laboral, trayectoria dentro de la empresa.• Podrá consultar reportes de una forma sencilla y práctica.• Control de las plantillas de forma automática estará controlando y auditando las plazas autorizadas, las ocupadas y las vacantes.• Capacidad de generar organigramas dinámicos.• Controlar estructuras corporativas.

Fuente: Diseño propio derivado del sistema de información que se desarrolló, 2013.

Para la validación del sistema se consideraron datos simulados del Departamento de Recursos Humanos de la Administración General de Servicios de Administración Tributaria, por lo que, los nombres de los empleados no son de trabajadores del SAT.

El sistema inicia con una interfaz de seguridad, para comprobar la autenticidad del usuario y verificar que el usuario cuente con la autorización y pueda hacer uso del, software con el objetivo de salvaguardar la información Figura 30.



Figura 30 Login. Fuente: Elaboración propia con datos del sistema de información desarrollado, 2013.

Este interfaz pide ingresar el Expediente y contraseña, donde usuario es el número de expediente y contraseña, se forma de cualquier cadena de caracteres alfanuméricos de máximo 8 dígitos, la cual es personal e intransferible.

Después de que el usuario autorizado ingreso, se muestra la pantalla principal, en ella se muestra una barra de menú en la parte superior, en la cual se enlistan las opciones (Catálogos, Control de Empleado, Reporte, Seguridad y Salir) cuenta con un menú amigable para el usuario, donde muestra una imagen de una carpeta, cada una de las pestañas ilustrativas, dan acceso a una opción del menú, como se muestra en la figura 31.

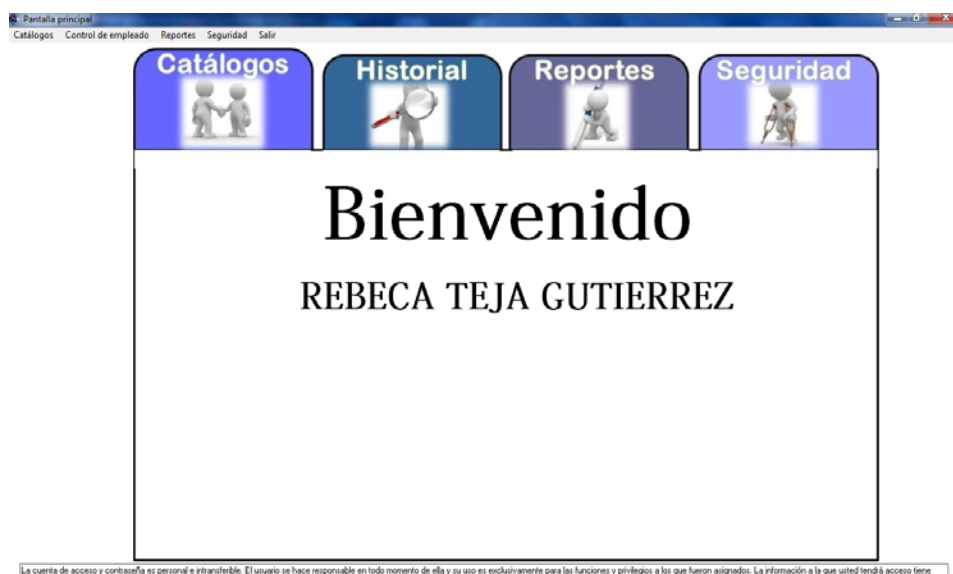


Figura 31 Pantalla principal. Fuente: Elaboración propia con datos del sistema de información desarrollado, 2013.

Al dar clic en la opción “CATÁLOGOS” se despliega un submenú con las opciones (Movimientos, Ubicación, Causa Baja, Área, Nivel, Rango) este submenú, se visualiza de dos formas, el tradicional desplegable y el ilustrativo, que tiene la misma funcionalidad (Figura 32), al dar clic en movimiento muestra una vista donde el usuario podrá capturar los diversos movimientos que puede tener un empleado durante su instancia en la empresa como son: Nuevo ingreso, Promoción, Permuta, Reubicación entre otros y permite asignarle un identificador para cada uno como se observa en la figura 33.

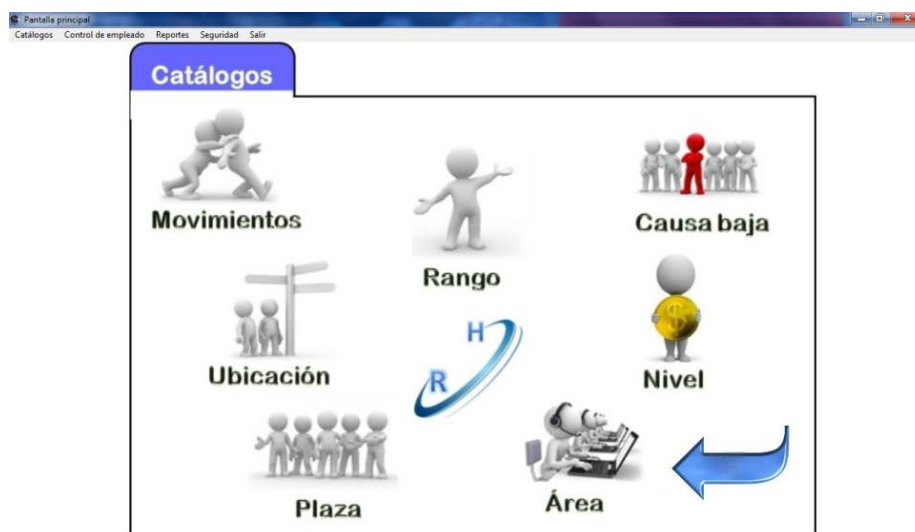


Figura 32 Catálogos. Fuente: Elaboración propia con datos del sistema de información desarrollado, 2013.



Figura 33 Catálogo de Movimientos. Fuente: Elaboración propia con datos del sistema de información desarrollado, 2013.

Con la opción Nivel de catálogos se muestra una vista donde el usuario podrá capturar lo niveles jerárquico que puede tener un empleado durante su instancia en la empresa como son: Administrador, Subadministrador, Jefe de Departamento, Enlace, entre otros y asignarle un identificador (Figura 34).

The 'Catálogo Nivel' window contains the following input fields:

- Nivel: 5
- Código nivel: A01807
- Sueldo: 1300
- Descripción: JEFE DE OFICINA
- Jerarquía: 0

Below the input fields is a table with the following data:

nivel	código n	sueldo	descrip	jerarqu
5	A01807	1300	JEFE DE OFICINA	0
2	CF04805	1000	SECRETARIA EJECUT	0
6	CF21810	140	AUDITOR FISCAL	0
7	CF21822	150	SUPERVISOR DE AUD	0
4	CF21838	120	AYUDANTE DE AUDIT	0
8	CF21859	160	COORDINADOR PROF	0
PQ3	CF21864	220	PROFESIONAL EJECU	0
PQ2	CF21865	210	PROFESIONAL DICTA	0

Figura 34 Catálogo de nivel. Fuente: Elaboración propia con datos del sistema de información desarrollado, 2013.

Con la opción Rango de catálogos muestra una vista donde el usuario podrá capturar el rango de los niveles permitidos en cada área (Figura 35).

The 'Catalogo de Rango' window contains the following input fields:

- Código rango: GC2012
- Rango: ADMINISTRADOR CENTRAL DE PLANEACION
- Rango inferior nivel: KA11
- Rango superior nivel: KA20
- Area: ADMINISTRACION CENTRAL DE PLANEACI

Below the input fields is a table with the following data:

c. rango	n. rango	r. infer. nivel	r. super. nivel	n. area
GC2012	ADMINISTRADOR CEN	KA11	KA20	ADMINISTRACION CEN
GC3078	ADMINISTRADOR DE	AA01	AA10	ADMINISTRACION CEN
GC3079	ADMINISTRADOR DE	AA01	AA10	ADMINISTRACION CEN
GC4080	SUBADMINISTRADOR	SB01	SB20	ADMINISTRACION CEN
GC4082	SUBADMINISTRADOR	SA01	SA20	ADMINISTRACION CEN
GC4083	SUBADMINISTRADOR	SA01	SA20	ADMINISTRACION CEN
GC4086	SUBADMINISTRADOR	SB01	SB20	ADMINISTRACION CEN

Figura 35 Catálogo de rango. Fuente: Elaboración propia con datos del sistema de información desarrollado, 2013.

Con la opción Ubicación de catálogos muestra una vista donde el usuario podrá capturar las características física del lugar donde se encuentra laborando el empleado así como: Módulo, Piso entre otros (Figura 36).

id ubicacion	n modulo	piso	ubicacion
101	3	0	090150D3PB
102	3	1	090150D301
103	3	2	090150D302
104	8	0	090150D8S1
105	8	0	090150D8PB
106	8	1	090150D801
107	8	2	090150D802

Figura 36 Catálogo de ubicación. Fuente: Elaboración propia con datos del sistema de información desarrollado, 2013.

Con la opción “Causa baja de catálogos” muestra una vista donde el usuario podrá capturar los diversos tipos de bajas que un empleado puede llegar a tener durante su instancia en la empresa como son: Renuncia, Abandono de empleo, Jubilación, Defunción, entre otros y asignarle un identificador (Figura 37).

id cbaja	n cbaja
101	DESPIDO
102	JUBILACION

Figura 37 Catálogo Causa de la baja. Fuente: Elaboración propia con datos del sistema de información desarrollado, 2013.

Con la opción Área de catálogos muestra una vista donde el usuario podrá capturar las áreas de las cuales lleva el control la empresa y asignarle un identificador (Figura 38).

N°	Nombre del área	Letra
101	ADMINISTRACION CENTRAL DE PLANEACION Y A	A
102	ADMINISTRACION CENTRAL DE FISCALIZACION B	B
103	ADMINISTRACION CENTRAL DE NORMATIVIDAD C	C
104	ADMINISTRACION CENTRAL DE NORMATIVIDAD D	D
105	ADMINISTRACION CENTRAL DE FISCALIZACION F	F

Figura 38 Catálogo área. Fuente: Elaboración propia con datos del sistema de información desarrollado, 2013.

Con la opción Control del empleado muestra una pantalla donde podrá dar de alta a un nuevo empleado, llenando con sus datos personales como: Nombre, RFC, CURP entre otros, además se definen los datos de la plaza que va a ocupar como son: Nivel, Régimen, Posición, Sueldo, Ubicación y Área y después dar clic en el icono agregar (Figura 39).

Historial del empleado

N_contrato	Movimiento	Inicio	Fin
67	PROMOCION	2012-10-18	
34	PERMUTA	2012-10-12	20
232	NUEVO INGRESO	2012-04-30	20

Figura 39 Control del empleado. Fuente: Elaboración propia con datos del sistema de información desarrollado, 2013.

Además permite modificar algunos de los datos que se requieran dando clic en el icono Modificar. Permite también realizar un nuevo contrato, Registrar una baja, Eliminar un registro y permite consultar el historial de cualquier empleado mediante el icono Historial y cuenta con el icono Excel que al darle clic puede exportar dicho reporte a un archivo .xls (Figura 40).

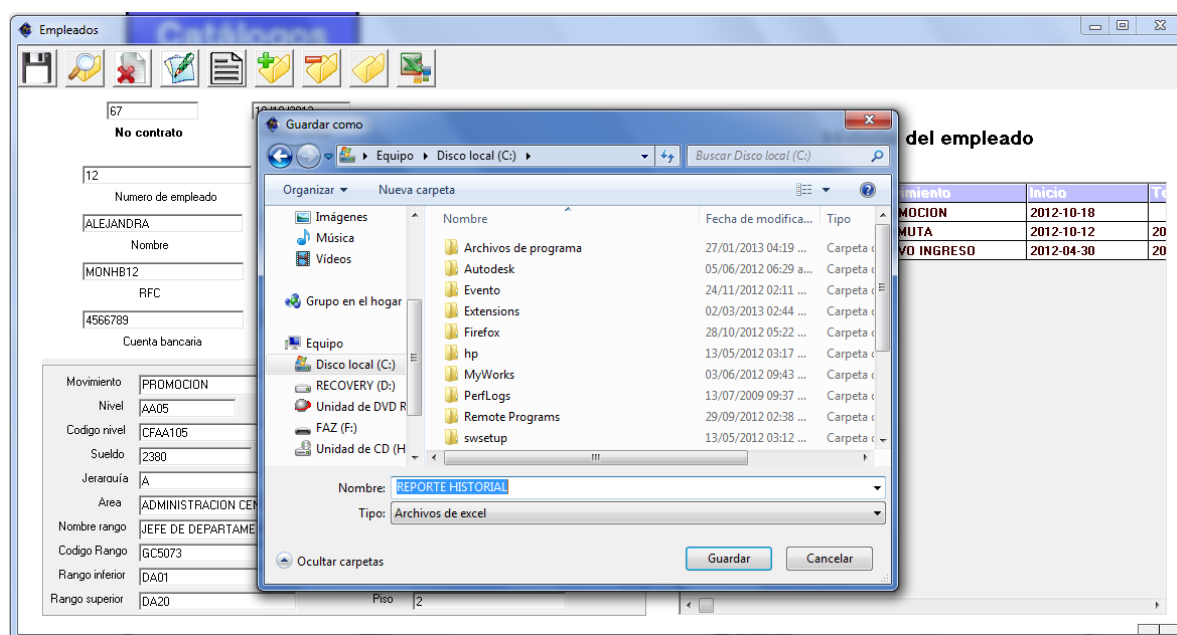


Figura 40 Exportar el historial del empleado a Excel. Fuente: Elaboración propia con datos del sistema de información desarrollado, 2013.

Al dar clic en la opción “REPORTES” despliega un submenú con las opciones (Movimiento, Bajas y Plaza) este submenú también se visualiza de dos formas, el tradicional desplegable y el ilustrativo, que tiene la misma funcionalidad (Figura 41), al dar clic en movimiento muestra una vista donde el usuario podrá visualizar algún reporte de los movimientos, ya sea de forma detalle o de resumen, el cual se realiza por Fecha, Área y Tipo de movimiento (Figura 42).



Figura 41 Reportes. Fuente: Elaboración propia con datos del sistema de información desarrollado, 2013.

Reporte movimiento

☒ Detalle
 ☐ Resumen
 ☐ Nombre completo

Movimiento:

Area:

Mes:
 Año:

Nombre_del_area	Número_empleado	Nombre
ADMINISTRACION CENTRAL DE PLANEACION Y PROGRAMACION	92722	JUAN
ADMINISTRACION CENTRAL DE PLANEACION Y PROGRAMACION	85711	SONI
ADMINISTRACION CENTRAL DE PLANEACION Y PROGRAMACION	77810	SANI
ADMINISTRACION CENTRAL DE PLANEACION Y PROGRAMACION	65432	EDU
ADMINISTRACION CENTRAL DE PLANEACION Y PROGRAMACION	112532	MAR
ADMINISTRACION CENTRAL DE NORMATIVIDAD	52781	FABI
ADMINISTRACION CENTRAL DE NORMATIVIDAD	89121	FER
ADMINISTRACION CENTRAL DE NORMATIVIDAD	10001	ROB
ADMINISTRACION CENTRAL DE NORMATIVIDAD	33574	ROB

Figura 42 Reportes de movimiento. Fuente: Elaboración propia con datos del sistema de información desarrollado, 2013.

Dicho reporte se podrá imprimir después de generar algún reporte o exportar en una hoja de Excel, mediante los icono de Imprimir o Excel respectivamente (Figura 43).

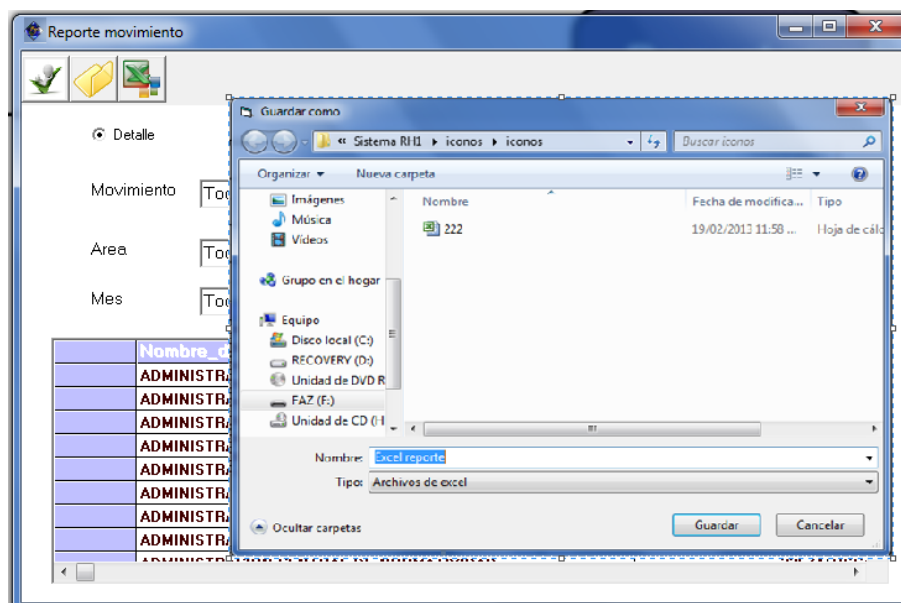


Figura 43 Generar registros a excel. Fuente: Elaboración propia con datos del sistema de información desarrollado, 2013.

En la opción baja de reportes muestra una vista donde el usuario podrá visualizar algún reporte de las bajas ya sea en forma de detalle o de resumen, el cual se realiza por Fecha, Área y Causa de la baja (Figura 44).

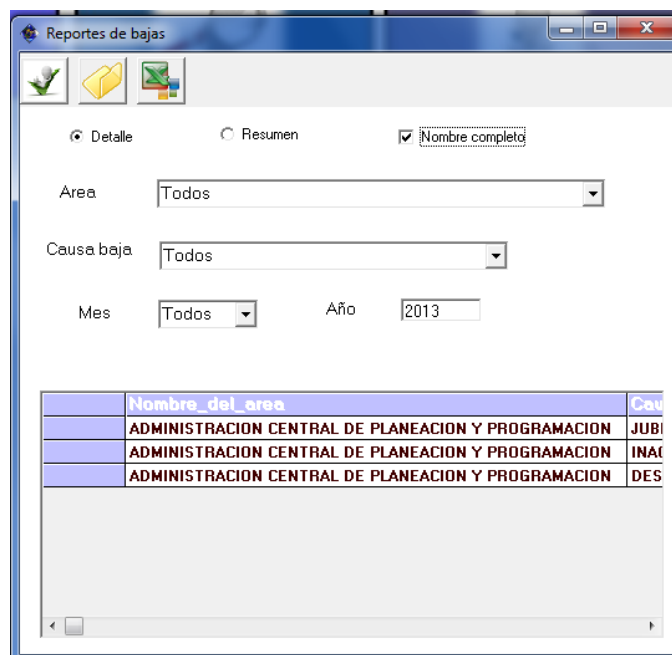


Figura 44 Reporte de bajas. Fuente: Elaboración propia con datos del sistema de información desarrollado, 2013.

Dicho reporte se podrá imprimir después de generar algún reporte o exportar en una hoja de Excel, mediante los icono de Imprimir o Excel respectivamente.

En la opción Seguridad, se podrá capturar a las personas autorizadas para tener acceso a la información del departamento, donde deberán ingresar Nombre, Expediente, RFC y Tipo de usuario (Figura 45). En tipo de usuario se podrá seleccionar la Categoría del empleado, es decir, cada Categoría tiene cierto privilegio por ejemplo: Categoría 1, el usuario que tenga este, tendrá control del sistema sin ninguna restricción, este usuario tambien podrá Agregar, Eliminar, Consultar, Reiniciar contraseña de los usuarios que estén registrados en el sistema.

n_empleado	nombre	paterno	materno	rfc	tipo_usuario
270	ADRIAN	TRUEBA	ESPINOSA	TRESA	Categoría 1
345	FABIOLA	ZAVALA	CRUZ	ZACRF	Categoría 2
3455	PEDRO	MONTEZ	CABRERA	MOCAP	Categoría 3

Categoría 1 - Administrador del sistema
Categoría 2 - Consultar y Agregar
Categoría 3 - Consultar














Figura 45 Seguridad. Fuente: Elaboración propia con datos del sistema de información desarrollado, 2013.

Por último la opción salir que permite cerrar completamente el sistema.

Para interactuar con el sistema se representan imágenes demostrando de manera general del funcionamiento de las interfaces en ejecución, para ello se agregaron iconos que hacen más amigable la interacción del usuario con el sistema, además,

describe la acción que realiza cada icono a fin de familiarizarse con ellos (Ver Tabla 28).

Tabla 28.- Iconos del sistema

Icono	Nombre	Función
	Seguridad	Permite o no el acceso al sistema.
	Eliminar	Da pie a la eliminación de un registro que esté presente en la base de datos en uso.
	Buscar	Da pie a la búsqueda de un registro por medio de una variable indicada.
	Agregar	Agrega los datos que se encuentran en la interfaz en un nuevo registro.
	Modificar	Modifica los datos registrados en la interfaz en uso.
	Reporte	Genera el reporte de acuerdo a las variables especificadas por el usuario.
	Exportar a Excel	Permite exportar un reporte en Excel.
	Nuevo contrato	Permite agregar un nuevo contrato, para ir formando el historial de un empleado.
	Guardar contrato	Da pie a guardar el nuevo contrato del empleado.
	Aplicar baja	Da pie a dar de baja un empleado.
	Reiniciar	Permitirá reiniciar la contraseña del usuario.
	Cancelar	Permite cancelar alguna acción.
	Salir	Da pie a cerrar el sistema.

Fuente: Elaboración propia con datos del sistema de información desarrollado, 2013.

VIII.- DISCUSIÓN.

Para determinar los procesos para la administración del personal y considerando las recomendaciones de [21] se tuvo contacto con el personal experto del departamento. Fijando la atención en los datos del personal, plazas que ocupan, áreas a las que pertenecen, movimientos que tienen en el trascurso laborado, datos de bajas de empleados y generación de reportes.

Una vez detectados los procesos, se identificaron las necesidades del departamento como lo indica [21], para asegurar la competitividad del departamento, se recopiló, clasificó y seleccionó la información necesaria, considerando las cualidades para que la información sea útil como dice [28]. Para desarrollar un sistema que proporcione la información clave que apoye a la toma de decisiones del departamento, como lo indica [24], además se asumieron los lineamientos y características para definir requerimientos que indican [14], [15], [38] y [37].

Ahora bien, se pasó a definir el tipo de sistema de información que se desarrollaría y se decidió que sería un Sistema de Información Administrativa (MIS) que es el que más se ajusta a las necesidades del departamento, según clasificación de [8] y [10] y se dedujo que el área de software indicada era la de software en tiempo real analizando lo que especifican [53] y [54].

El modelo del sistema de acuerdo a los requerimientos se empleó el UML según [61] y [62] y se estructuraron los diagramas de casos de uso de acuerdo a [72], [60], [57] y [95].

Para desarrollar el sistema se consideró el modelo en espiral ya que resultó una mejor alternativa, ya que permite la implementación de prototipos con un ciclo de vida indefinido, con ello se permite realizar mejoras y adiciones al prototipo propuesto tal como lo mencionan [53], [54] y [71]. Además de acuerdo a [54], se

tuvo una evaluación del riesgo, con ello se resolvieron varias problemáticas que se fueron presentando al desarrollar el sistema.

El sistema desarrollado en este trabajo está implementado a la medida del departamento y una de sus mejores aportaciones es la generación de reportes, ya que con ellos se reduce tiempo, con lo cual los directivos pueden tomar decisiones de forma inmediata.

Los reportes están divididos en tres categorías: son Movimientos, Bajas y Plazas. Figura 46.



Figura 46 Generación de reportes Fuente: Elaboración propia con datos del sistema de información desarrollado, 2013.

En la categoría de movimientos se tiene 4 variantes: tipo de movimiento, el área, mes y año. Lo que ayuda en gran medida, ya que se realizan indicadores cada mes, donde se debe indicar cuántos movimientos se realizaron en determinado periodo por cada tipo de movimiento, además se entregan también estadísticas de cada una de las 12 áreas, se sabe cuántos movimientos se tuvo y de qué tipo por lo cual las 4 variantes facilitan la selección de los campos para los indicadores que se pidan.

Los tiempos que se ahorran son considerables, por ejemplo al hacerlo manualmente para poder generar los indicadores de cada mes se necesita revisar: un archivo, donde se capturan los movimientos realizados, el tipo de movimiento, la fecha en la que se realiza y el área que lo solicita. Para lograrlo se deben hacer varios filtros en una hoja de cálculo donde están los datos capturados, al final se obtiene un resumen y con ello adaptar la información para poder visualizarla en una gráfica, todo este proceso es tardado, aproximadamente se realiza en hora y media con una certeza del 90% de confiabilidad ya que se corre el riesgo de considerar datos mal filtrados.

Con el sistema desarrollado se puede rápidamente seleccionar el periodo que se requiera por medio de opciones en la pantalla, se pueden obtener en dos formas; a detalle donde se muestran los datos del movimiento con los nombres de las personas involucradas en los movimientos entre otros datos relacionados y en forma de resumen; donde se muestra la cantidad exacta de los movimientos, lo que permite la traficación de inmediato. Estas consultas se hacen en 2 min, la figura 47 expone un reporte con movimientos a detalle y la figura 48 muestra el reporte a nivel de resumen.

Reporte de movimientos

Movimiento: NUEVO INGRESO
 Area: Todos
 Mes: Todos

Número empleado	nombre
2030	FABIOLA ZAVALA CRUZ
2031	STEPHANIE CASTILLO CAMACHO
2032	JUAN CARLOS ARANZABUENA
2034	MIGUEL ANGEL RAMIREZ
2033	CIRENIA CRUZ MORALES
2035	MARCO ANTONIO ZARATE
2036	JUAN GABRIEL CASTILLO
2037	REBECA TEJERA GUTIERREZ
2041	ADRIAN TORRES

Reporte de movimientos

Zoom: 100%

Administración General de Grandes Contribuyentes
 Subadministración de Enlace Administrativo
 Departamento de Recursos Humanos

Mes: Todos Año: 2013

NOMBRE DEL AREA	N	NOMBRE	POBICION	MOVIMIENTO	NIVEL	COD PUESTO	REGIMEN	F DE INICIO
ADMINISTRACION CENTRAL DE PLANEACION Y PROGRAMACION	2030	FABIOLA	1030010	NUEVO INGRESO	CP2188	GC2012	PRESUPUESTO	07/02/2013
ADMINISTRACION CENTRAL DE PLANEACION Y PROGRAMACION	2031	STEPHANIE	1030060	NUEVO INGRESO	CPKA101	GC4080	PRESUPUESTO	16/02/2013
ADMINISTRACION CENTRAL DE PLANEACION Y PROGRAMACION	2032	JUAN CARLOS	1030062	NUEVO INGRESO	CP58101	GC4082	PRESUPUESTO	28/03/2013
ADMINISTRACION CENTRAL DE PLANEACION Y PROGRAMACION	2034	MIGUEL ANGEL	1030067	NUEVO INGRESO	CPDA111	GC3079	PRESUPUESTO	26/04/2013
ADMINISTRACION CENTRAL DE PLANEACION Y PROGRAMACION	2033	CIRENIA	1030068	NUEVO INGRESO	CPPC003	GC4086	PRESUPUESTO	14/04/2013
ADMINISTRACION CENTRAL DE PLANEACION Y PROGRAMACION	2035	MARCO ANTONIO	1030072	NUEVO INGRESO	CPDA111	GC4083	PRESUPUESTO	27/04/2013
ADMINISTRACION CENTRAL DE PLANEACION Y PROGRAMACION	2036	JUAN GABRIEL	1030088	NUEVO INGRESO	CP21909	GC4080	PRESUPUESTO	19/04/2013
ADMINISTRACION CENTRAL DE PLANEACION Y PROGRAMACION	2037	REBECA	1030100	NUEVO INGRESO	CP58109	GC4072	PRESUPUESTO	13/04/2013
ADMINISTRACION CENTRAL DE DIVERSOS	2041	ADRIAN	1030152	NUEVO INGRESO	CPDA105	GC5040	PRESUPUESTO	28/06/2013

Páginas: 1

Figura 47 Reportes de movimientos a nivel Detalle Fuente: Elaboración propia con datos del sistema de información desarrollado, 2013.

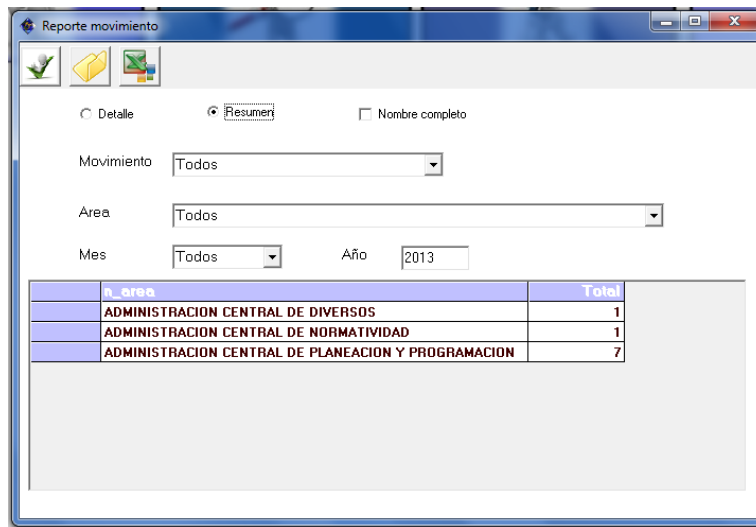


Figura 48 Reportes de movimientos a nivel Resumen Fuente: Elaboración propia con datos del sistema de información desarrollado, 2013.

La categoría de bajas también está compuesta por 4 variantes estas son área, causa de la baja, mes y año. Es aquí donde se generan reportes de las bajas de personal por determinado periodo, cuantas bajas tuvo cada área, cuantas bajas en general de todas las áreas o cuantas bajas dependiendo su tipo. Las bajas se deben ver reflejas en los indicadores para saber qué área tuvo más bajas. Estos indicadores se realizan cada mes o cada que un administrador requiere la información. Para poder elaborar este reporte se requiere filtrar datos de un archivo, donde se van capturando las bajas y después organizar la información para posteriormente graficarla, el proceso es tarda aproximadamente 2 horas y la confiabilidad es de 90%. En cambio con el sistema se realiza en 2 min y con la certeza de un 100% de confiabilidad. Esto se puede ver en la figura 49 y 50.

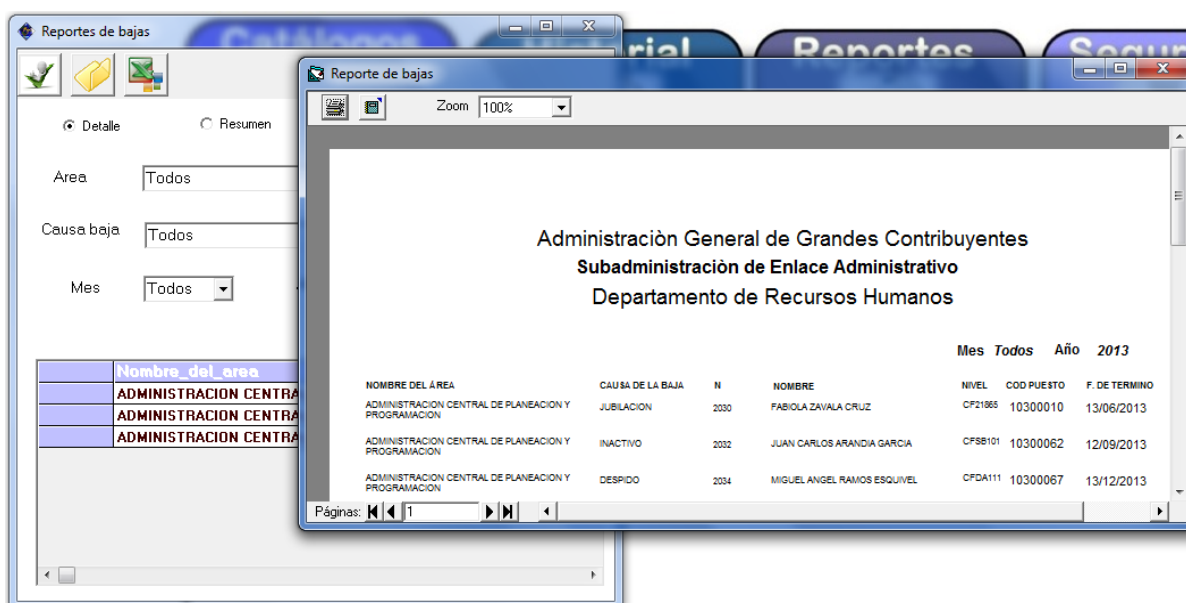


Figura 49 Reportes de bajas a nivel Detalle Fuente: Elaboración propia con datos del sistema de información desarrollado, 2013.

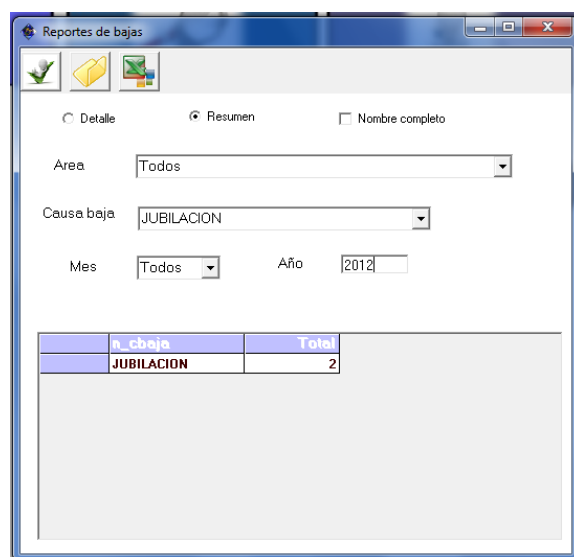


Figura 50 Reportes de bajas a nivel Resumen Fuente: Elaboración propia con datos del sistema de información desarrollado, 2013.

Hay otro reporte llamado Plazas, es uno de los más utilizados, ya que la información se requiere lo más pronto posible, con este reporte se toman decisiones para la contratación de personal. Con él se puede llevar un control de la cantidad de personal que se encuentra laborando, el nivel que se ocupa jerárquicamente, el tipo de plaza que tienen y con cuántas vacantes cuenta la institución o cuántas

plazas tiene cada área. Para hacer manualmente este reporte, se requiere revisar un catálogo de plazas, revisar un archivo de vacantes, otro archivo de contratación para que con la unión o cruce de esos tres archivos, se pueda generar el reporte solicitado, esto es tedioso, y se realiza bajo presión, aproximadamente se llevan 2 a 3 horas para generarlo, se corre el riesgo de error por la presión de entregarlo de forma inmediata. Con el sistema se realiza solo con la selección de las variantes sin tener que estar revisando varios archivos y se genera en aproximadamente en 2 minutos, la figura 51 y 52 muestran los datos obtenidos.

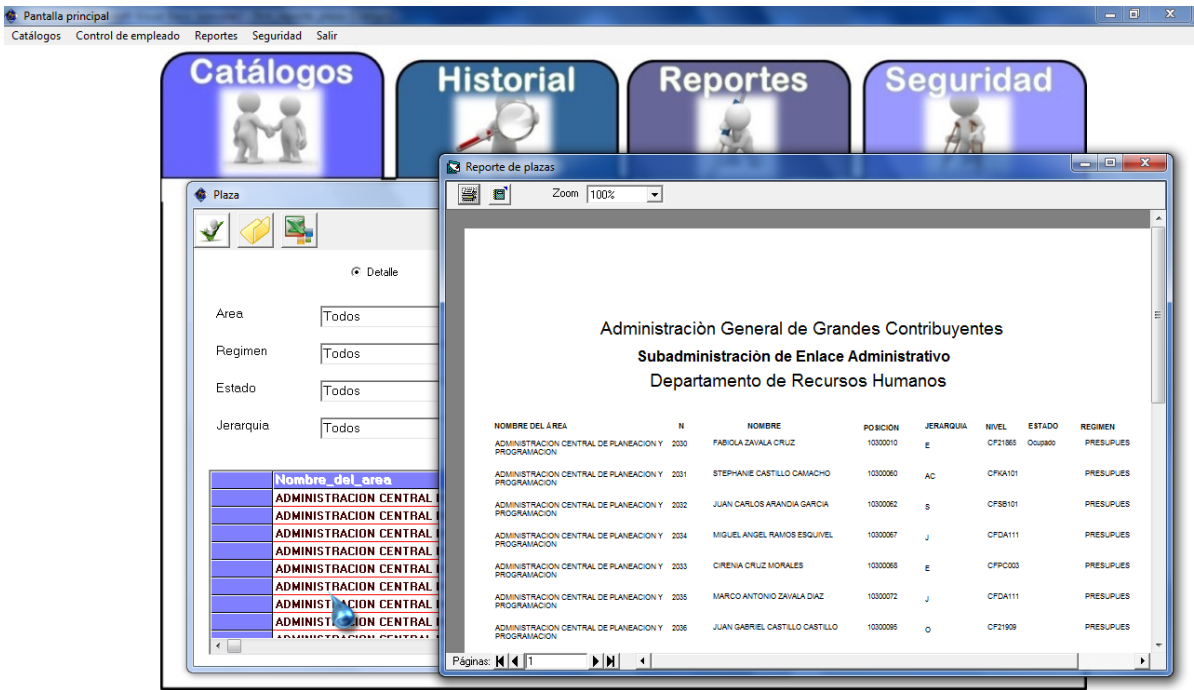


Figura 51 Reportes de plaza a nivel Detalle Fuente: Elaboración propia con datos del sistema de información desarrollado, 2013.

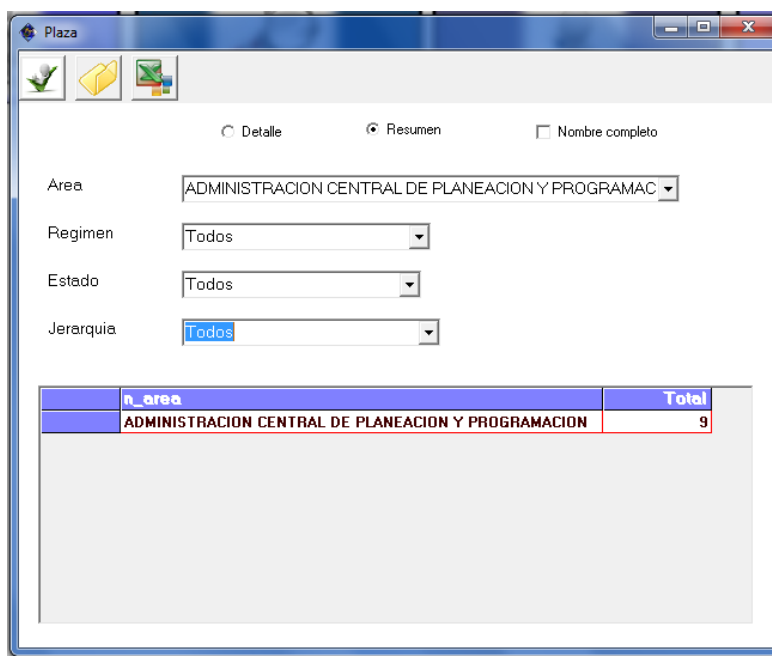


Figura 52 Reportes de plaza a nivel Resumen Fuente: Elaboración propia con datos del sistema de información desarrollado, 2013.

Otro reporte es el Historial del Empleado, donde se conoce la antigüedad, esta información al sacarla de forma manual se obtiene en aproximadamente 3 horas, debido a la revisión manual en todas las carpetas clasificadas por año. Esta es una de las actividades más pesadas del departamento, ya que se complica cuando se requiere información de varios empleados. Con el sistema solo se teclea el número del empleado y se obtiene el reporte en cuestión de segundos, como se muestra en la figura 53.

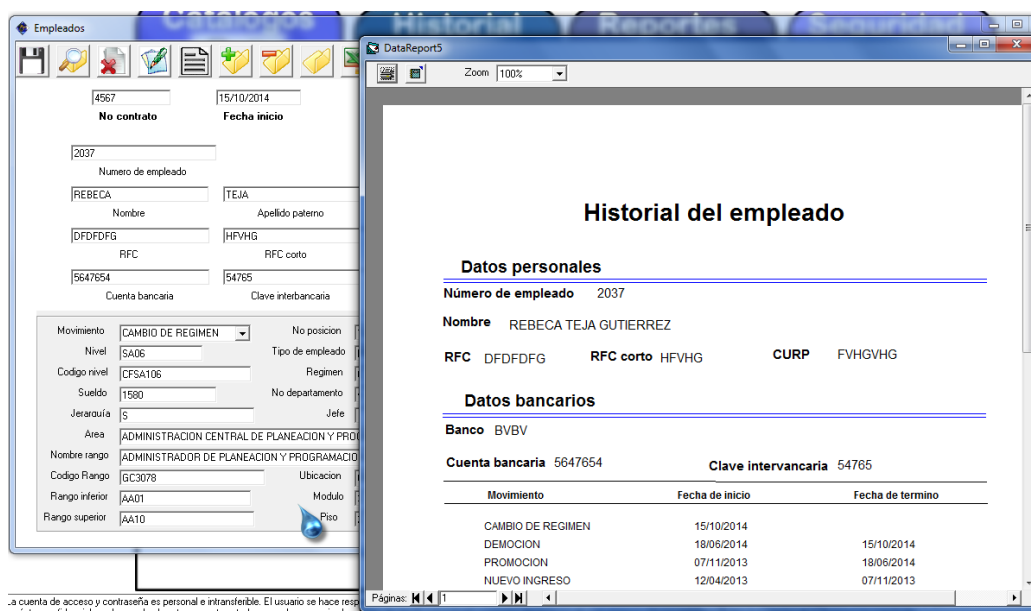


Figura 53 Historial del empleado Fuente: Elaboración propia con datos del sistema de información desarrollado, 2013.

El sistema permite exportar la información a Excel para manipular la información, en cuadros, gráficas y otros documentos que se requieran, como se muestra en la Figura 54.

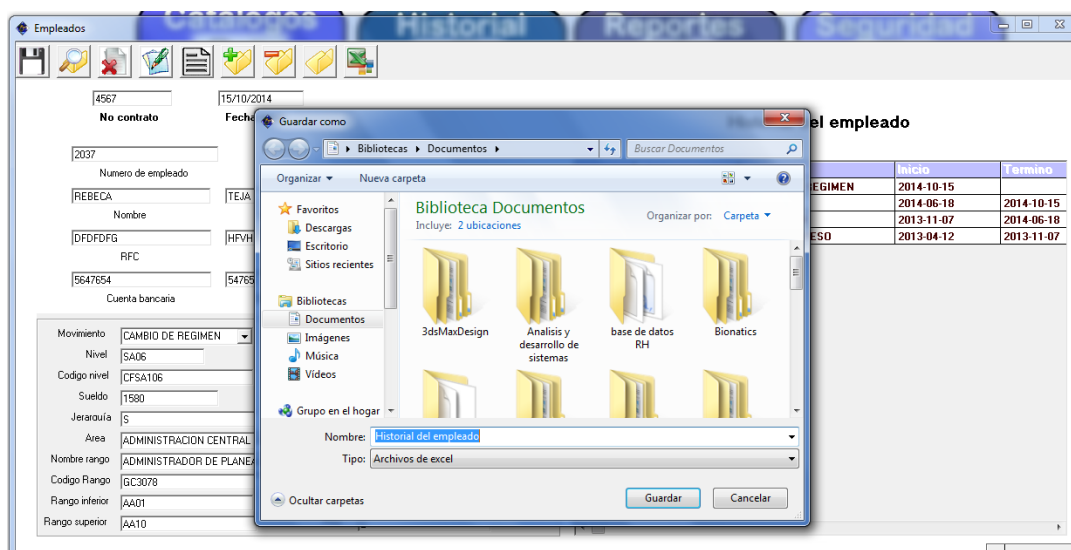


Figura 54 Exportación de la información a Excel Fuente: Elaboración propia con datos del sistema de información desarrollado, 2013.

Los reportes presentados son la parte fundamental del sistema, ya que permite obtener información de forma inmediata y con ello se ahorra tiempo y se tiene la certeza de que la información está libre de errores.

En el mercado existen sistemas que manejan la administración de los recursos humanos uno de ellos es el **Isocron Systems**, este permite acceder a la información de los empleados rápidamente, pero no permite la generación de reportes, lo que ocasiona una dificultad para la toma de decisiones en tiempo y forma. Otro sistema comercial es el **Revolución** y cuenta con una interfaz fácil con menús ágiles, que ayudan al proceso de interactuar entre el humano y sistema, pero deja a un lado la generación de reportes al igual que el Isocron Systems.

También está el llamado **Sistema FortiaRH**, está dividido por módulos, la captura de datos es amplia, está bajo la tecnología Microsoft .NET, tiene la desventaja que el mantenimiento es en múltiples lenguajes. Mantener un proyecto en múltiples lenguajes es costoso. Si una aplicación está realizada en varios lenguajes se necesitan expertos en varios lenguajes para entenderla y mantenerla, además .NET no es multiplataforma. La plataforma .NET sólo está disponible para la familia Windows y en cuanto a licencias es un código cerrado, no hay licencias libres.

El **SerSoft**, es otro sistema comercial para recursos humanos y cuenta con una buena base de datos desarrollada en MySQL donde la administración del personal ocupa el lugar central de la aplicación pero dejan a un lado la interfaz, no cuenta con menús fáciles de utilizar y dificulta la captura de los datos.

El sistema de administración de recursos humanos desarrollado en este trabajo, tiene ventajas competitivas con relación a los sistemas antes mencionados debido a que cuenta con una de las mejores características de cada uno de ellos, en un solo sistema. Tiene una interfaz amigable con menús fáciles de usar, los formularios están diseñados para que la información se capture lo más rápidamente posible y permite la generación de reportes que son indispensables para la toma de

decisiones, así como, la consulta del historial de cualquier empleado en tiempo real y permitiendo tener actualizada la estructura.

Cuenta también con una base de datos desarrollada en MySQL, tiene la ventaja de que es una plataforma de trabajo bastante sencilla de utilizar, cuenta con una licencia de funcionamiento GPL (Licencia Pública General). Además permite acceder a todos los campos que resguardan los datos de trabajo, puede ingresar una enorme cantidad de datos por columna de trabajo. Soportar hasta 32 índices de tablas diferentes y capacidad de realizar pruebas de rendimiento en tiempo real, este gestor cuenta con una gran estabilidad. MySQL es un gestor de base de datos relacional que es compatible con otros lenguajes de programación los cuales son PHP, C y HTML entre otros **[76]**.

IX.- CONCLUSIÓN

Con el sistema desarrollado se reduce el tiempo de respuesta en la entrega de reportes para la administración de personal del SAT.

La capacidad de realizar reportes para la administración del personal es una cualidad que hace destacar el software desarrollado, en contraste con los comerciales que se encontraron.

El sistema propuesto reduce costos al disminuir los tiempos para la contratación de personal y hacer la gestión de la administración más eficiente.

La interface del sistema propuesto estimula que sea utilizado por su fácil manejo y entendimiento para ser operado.

La interface amigable por medio de menús de fácil acceso permite la interacción con usuarios sin conocimientos técnicos al sistema.

Se aumenta en un 100% la confiabilidad de la información que se refleja en los reportes, por lo que la toma de decisiones tiene un alto grado de certeza.

Con los reportes implementados se disminuye el tiempo de respuesta en veinte minutos en movimientos, una hora con cincuenta y siete minutos en reportes de bajas, y dos horas con cincuenta y siete minutos en reportes de plazas.

Se ganan tres horas al obtener el historial de cualquier empleado, sin margen de error.

X.- RECOMENDACIONES

Se recomienda diseñar y programar un módulo de generación de estructura organizacional, haciendo uso de la información almacenada en la base de datos, que permita visualizar el organigrama del departamento de acuerdo a los movimientos capturados, permitiendo así la detección de los movimientos de una forma más sencilla y rápida.

Es conveniente desarrollar un módulo de graficación para dejar de depender del Excel en un 100%.

XII.- BIBLIOGRAFÍA

[1] David A. de Canzo; Stephen P. Robbins (2006) ADMINISTRACIÓN DE RECURSOS HUMANOS LIMUSA, México, Pág.5, 6.

[2] Gary Dessler (2001) ADMINISTRACIÓN DE PERSONAL (8° Edición) Prentice Hall, Pág.2, 3.

[3] William B. Werther, Jr; Keith David (2004) ADMINISTRACIÓN DE PERSONAL Y RECURSOS HUMANOS (5° Edición) Mc Graw Hill, España, Pág. 9-14.

[4] George Bohlander ; Scott Snell; Arthur Sherman (2001) ADMINISTRACIÓN DE RECURSOS HUMANOS (10° edición) Thomson learning, México, Pág. 4-17.

[5] William B. Werther ; Keith Davis (2008) ADMINISTRACIÓN DE RECURSOS HUMANOS, EL CAPITAL HUMANO DE LAS EMPRESAS (6° Edición) Mc Graw Hill, España, Pág. 5-25.

[6] Silvia Pérez Campuzano (1999) GLOSARIO DE ADMINISTRACIÓN DE PERSONAL. (1° Edición) IAPEM, Estado de México Pág. 27 -141.

[7] Miguel Castaño (2008) FUNDAMENTOS Y MODELOS DE BASES DE DATOS (2° Edición) Alfaomega ra-ma E.U.A., Pág. 11-25.

[8] James A. Senn, (2007) ANÁLISIS Y DISEÑO DE SISTEMAS DE INFORMACIÓN (2° Ed.) Mc Graw Hill México, Pág. 19- 49.

[9] Michael V. Mannino, (2007) ADMINISTRACIÓN DE BASES DE DATOS DISEÑO Y DESARROLLO DE APLICACIONES (3ra Ed.) Mc Graw Hill México Pág. 23,24, 25, 26, 80 y 81.

[10] Jeffrey L. Witten y Lonnie D. Bentley, (2008) ANALISIS DE SISTEMAS DISEÑO Y METODOS (7° Ed.) Mc Graw Hill México, Pág. 24 – 29.

[11] Mario Gerardo Piattini Velthuis, (2007) ANALISIS Y DISEÑO DETALLADO DE APLICACIONES INFORMATICAS DE GESTIÓN (1 Edición) ra-ma, Pág. 11-25.

[12] Ian Sommerville (2005) INGENIERÍA DEL SOFTWARE (7° edición) Pearson Addison Wesley, Madrid España, Pág. 119-151.

[13] Kenneth E. Kendall, Julie E. Kendall (2005) ANALISIS Y DISEÑO DE SISTEMAS (6° Edición) Pearson Prentice Hall, México, Pág. 89-97.

[14] Piattini Velthuis, Mario G. Peso Navarro, Emilio (2008) AUDITORIA DE TECNOLOGÍAS Y SISTEMAS DE INFORMACIÓN (1° Edición) Alfaomega rama.

[15] Gómez Vieites Alvaro (2010) SISTEMAS DE INFORMACIÓN HERRAMIENTAS PRACTICAS PARA LA GESTIÓN (3° Edición) Alfaomega rama.

[16] http://www-2.dc.uba.ar/materias/isoft1/2001_2/apuntes/CasosDeUso.pdf

[17] Francia Celis de Soto (2006) LA GESTION DE RECURSOS HUMANOS EN LAS ORGANIZACIONES DE SERVICIO .

[18] Cassandra Carrido Trejo (2007) LA EDUCACIÓN EN LA TEORIA DEL CAPITAL HUMANO Y EL OTRO.

[19] De Haro García José Manuel (2003) GESTIONANDO EL VALOR DE LA FUNCIÓN DE LOS RECURSOS HUMANOS Vol.19 num.3 Madrid España.

[20] Muñoz Pereira Claudio (2006) LOS RECURSOS HUMANOS Y SU IMPORTANCIA PARA EL DESARROLLO SECTORIAL num.6 Valdivia Chile.

[21] Camejo C Armando J., Cajas Magda RESPONSABILIDAD SOCIAL: FACTOR CLAVE DE LA GESTION DE LOS RECURSOS HUMANOS EN LAS ORGANIZACIONES DEL SIGLO XXI (2009) Núm. 21 España.

[22] Valencia Rodríguez Marino (2005) EL CAPITAL HUMANO OTRO ACTIVO DE SU EMPRESA Vol.1 Núm. 2 Cali, Colombia.

[23] Gerón Piñón Gabriela (2003) LOS SISTEMAS DE INFORMACIÓN EN EL ARRANQUE DEL NEGOCIO, México.

[24] Delegación costa rica (2002) COSTA RICA: SISTEMA DE INFORMACIÓN DE RECURSOS HUMANOS, México.

[25] Área de consultoría Organizacional (2002) SISTEMA DE INFORMACIÓN DE RECURSOS HUMANOS, México.

[26] López Escalera Rigoberto (2009) PARAMETROS DE LOS SISTEMAS, Morelia Michoacán.

[27] Monroy Alvarado German Sergio (2000) SISTEMAS DE INFORMACIÓN ADMINISTRATIVA, México DF.

[28] García RM (2007) LA IMPORTANCIA DE LA ADMINITRACIÓN DE LA INFORMACIÓN Y SU EFECTO LABORAL Puebla, México.

[29] García de Hurtado María C.(2008) EVOLUCION HISTORICA DEL FACTOR HUMANO EN LAS ORGANIZACIONES DE RECURSO HUMANOS A CAPITAL INTELECTUAL, Venezuela.

[30] Cutro Alfonso, Soto Sergio, Mariño Sonia y Valesani María E (2009) DISEÑO DE UN SISTEMA DE INFORMACIÓN PARA LA GESTIÓN Y ADMINISTRACIÓN DE CURSOS. MODELO ELABORADO A PARTIR DE LAS JORNADAS DE FORMACIÓN Y ACTUALIZACIÓN EN INFORMÁTICA (FACENA). México.

[31] <http://www.isocron.net/soluciones/recursos-humanos>

[32] <http://www.fortia.com.mx/index.php/soporte.html>

[33] <http://www.sersoft.com.mx/rrhh.html>

[34] <http://revolutionenlanube.com/Demo.aspx>

[35] Sánchez Egbert (2005) LOS SISTEMAS DE INFORMACIÓN Y LOS PRINCIPALES ACTORES UNA APROXIMACIÓN RAZÓN Y PALABRA Vol.10 Num44 Edo, de México , México.

[36] Dr. Donald Klingner (2003) LA ADMINISTRACIÓN DE RECURSOS HUMANOS Y LA MODERNIZACIÓN DEL ESTADO University of Colorado Panamá.

[37] Arias Chávez Michael (2005) INGENIERÍA DE REQUERIMIENTOS Y SU IMPORTANCIA EN EL DESARROLLO DE PROYECTOS DE SOFTWARE, Costa Rica.

[38] Gómez Fuentes María del Carmen (2011) ANÁLISIS DE REQUERIMIENTOS UAM, México.

[39] Frederik P. Brooks (1987) LA INGENIERÍA DEL SOFTWARE. IEEE Computer.

[40] Ricardo González Lozano (2010) BASE DE DATOS CON ACCESS 2007, Alfaomega, México.

- [41] Ramez shamkant B. FUNDAMENTOS DE SISTEMAS DE BASE DE DATOS
- [42] I T Hawriyszkiewicz, (2000) ANÁLISIS Y DISEÑO DE BASE DE DATOS, LIMUSA Noriega editores, México.
- [43] Rosales Suárez Eloisa INTRODUCCIÓN AL DISEÑO DE BASE DE DATOS
- [44] Rob Peter/Coronel Carlos, (2004) SISTEMAS DE BASES DE DATOS DISEÑO, IMPLEMENTACIÓN Y ADMINISTRACIÓN Thomsom.
- [45] Marques Mercedes, BASES DE DATOS
- [46] Sorli Rojo Ángela, Merlo Vega Jose Antonio, BASE DE DATOS DE INFORMACION Y DOCUMENTACION EN INTERNET
- [47] Sánchez Jorge, PRINCIPIOS SOBRE BASES DE DATOS BASE DE DATOS RELACIONALES
- [48] Abram silberschatz, Henry F. Korth, S. sudarchan, FUNDAMENTOS DE BASES DE DATOS 4TA EDICION Mc Graw hill
- [49] Ramakristnan Rashu, Gherke Johannes (2007), SISTEMAS DE GESTION DE BASES DE DATOS McGrawHill
- [50] http://www.ecured.cu/index.php/Base_de_datos_relacional
- [51] http://es.wikipedia.org/wiki/Sistema_de_gesti%C3%B3n_de_bases_de_datos
- [52] Mariño, Sonia I.; Godoy, María Viviana; Petris, Raquel (2010) CONCEPTOS DIALÉCTICOS REFLEJADOS EN UN PROCESO DE INGENIERÍA DEL

SOFTWARE. ANÁLISIS PRELIMINAR vol. 10, núm. 3 Universidad del Zulia,
Venezuela

[53] Olguín Espinoza José Martín (2004) ANÁLISIS ORIENTADO A OBJETOS INGENIERÍA DEL SOFTWARE, México.

[54] Galo Fariño R. (2011) MODELO ESPIRAL DE UN PROYECTO DE DESARROLLO DE SOFTWARE ADMINISTRACIÓN Y EVALUACIÓN DE PROYECTOS, Universidad Estatal De Milagro 8vo Semestre.

[55] Corcos Daniel (2000) EL MODELO ESPIRAL ITBA Reportes Técnicos CAPIS

[56] Zapata, Carlos Mario; González, Guillermo (2008) ESPECIFICACIÓN FORMAL EN OCL DE REGLAS DE CONSISTENCIA ENTRE LOS DIAGRAMAS DE CLASES Y CASOS DE USO DE UML Y EL MODELO DE INTERFACES Vol. 7, Núm. 12 Universidad de Medellín Colombia.

[57] Tabares, Marta Silvia; Pineda, Juan Diego; Barrera, Andrés Felipe (2008) UN PATRÓN DE INTERACCIÓN ENTRE DIAGRAMAS DE ACTIVIDADES UML Y SISTEMAS WORKFLOW Núm. 10, Colombia

[58] Fernández Luis, J. Lara Pedro (2001) GENERACIÓN DE CASOS DE PRUEBA A PARTIR DE ESPECIFICACIONES UML Universidad Europea de Madrid.

[59] Rodríguez, Moisés; Verdugo, Javier; Coloma, Ramón; Genero, Marcela; Piattini, Mario (2010) METODOLOGÍA PARA LA EVALUACIÓN DE LA CALIDAD EN LOS MODELOS UML Vol. 6, Núm. 1, Asociación de Técnicos de Informática España.

[60] Roxana S. Giandin, Claudia F. Pons (1900) RELACIONES ENTRE CASOS DE USO EN EL UNIFIED MODELING LANGUAGE revista colombiana de computación Volumen 1, número 1. La Plata, Argentina.

[61] Hernández Orallo Enrique (2003) EL LENGUAJE UNIFICADO DE MODELADO basado en Jacobson, G. Booch, J. Rumbaugh (2000) EL PROCESO UNIFICADO DE DESARROLLO, Addison Wesley

[62] Gutiérrez Lázaro Juan Carlos (2009) DIAGRAMAS DE CLASES Y CASOS DE USO Dep. Ingeniería del Software e Inteligencia Artificial Facultad de Informática Universidad Complutense Madrid

[63] Frederik P. Brooks (1987) LA INGENIERÍA DEL SOFTWARE. IEEE Computer.

[64] Peter Rob / Carlos coronel (2004), SISTEMAS DE BASE DE DATOS DISEÑO IMPLEMENTACIÓN Y ADMINISTRACIÓN, Thomson, México

[65] Ricardo González Lozano(2010). BASE DE DATOS CON ACCESS 2007, Alfaomega, México,

[66] <http://www.webyempresas.com/la-clausula-where-en-sql/>

[67] <http://www.softredmania.com/sena/sql.pdf>

[68] <http://es.scribd.com/doc/64278624/SQL-Lenguaje-Estructurado-de-Consultas>

[69]http://cvb.ehu.es/open_course_ware/castellano/social_juri/herrami_gestion/cursosql.pdf

[70] Javier Neyra Bravo (2011), ADMINISTRACIÓN DE DATOS EN SQL, Editorial Mc Graw hill, México

[71] Roger S. Pressman (2002), INGENIERIA DE SOFTWARE un enfoque práctico, Mc Graw Hill (Quinta Edición), Madrid.

[72] Santiago Seria (2000), INGENIERIA I CASOS DE USO –Un Método Práctico para Explorar Requerimientos

[73] <http://www.definicionabc.com/tecnologia/mysql.php>

[74] <http://www.identi.li/index.php?topic=48822>

[75] Luis Joyanes Aguilar / Antonio Muñoz Clemente (2000) MICROSOFT VISUAL BASIC 6.0, Mc Graw Hill (Primer Edición), Madrid.

[76] Cesar Pérez (2008) MySQL PARA WINDOWS Y LINUX, Alfaomega (Segunda edición), México

[77] Juan Pavón Mestras (2004) FACULTAD DE INFORMÁTICA UCM
<http://www.vc.ehu.es/jiwotvim/ISOFT2008-2009/Teoria/Bloquel/Transp-03Proceso-Pressman.pdf>

XIII. ANEXOS

¿Cuál es el giro de su empresa?

El giro de la empresa es de servicio

¿Cuál es el objetivo principal de su área?

Llevar el control del personal de la Administración General que nos compete. La cual se constituye por 12 administraciones centrales.

¿A cuántos empleados quieren controlar en su empresa?

Estamos hablando de más de 1000 empleados.

¿Qué información requieren controlar del empleado?

Nombre, Tipo de plaza que ocupa, Nivel, Sueldo, RFC, CURP, Cuenta bancaria, Fecha de ingreso, fecha de baja, administración a la que pertenece, numero de empleado, etc....

¿Qué tipo de movimientos tienen los empleados?

- Permuta (cuando un empleado cambia plaza con otro empleado)
- Promoción (cuando un empleado sube de nivel)
- Democión (cuando un empleado baja de nivel)
- Cambio de adscripción (cuando un empleado cambia de administración)

¿Qué actividades realizan?

- Llevar registro de empleados, esto implica control de nuevos ingresos, bajas, movimiento, ubicación, plazas ocupadas y vacantes. Llevar la actualizada la estructura organizacional de cada una de las administraciones.
- También se realizan entrega de reportes mensuales y anuales de todos los movimientos de los que se lleva el control.
- Requisar los formatos establecidos para realizar trámites de la administración.

¿Cuáles la problemática a la que se enfrentan para lograr ese objetivo?

- Nos tardamos en buscar algún dato de un empleado en específico.
- Se nos dificulta recabar la información de todos los empleados y movimiento que tuvieron para la entrega de reporte que se realiza cada mes y cada año.
- Nos tardamos en investigar el historial del empleado, saber desde cuando entro que movimiento y plazas tuvo hasta la baja si es el caso.
- Dificultad en la autenticidad y actualización de la estructura organizacional de las doce administraciones.

¿Qué aplicaciones utilizan para realizar sus actividades?

Word para requisitar formatos, Excel almacenamos toda la información que se genera en la administración y graficas de los reportes, PowerPoint para exponer los reportes mensuales y anuales y Visio en el cual se utiliza para la elaboración de la estructura organizacional.

¿Cuentan con un sistema?

Si, existe un sistema externo pero no se cuenta con un sistema interno de nuestra administración general.

¿Les gustaría contar con un sistema que facilite sus procesos?

Si

¿Qué les gustaría que hiciera el sistema?

Que nos facilite la obtención de la información de lo empleado que elaboran en la administración, nos facilite en sacar los reportes que nos solicitan y llevar un mejor control y autenticidad en la información.

¿Cómo les gustaría que fuera?

Que sea fácil de usar y rápido.